84000440

CONTROL DATA CORPORATION

CONTROL DATA®
USAGE ACCOUNTING UTILITY
REFERENCE MANUAL

CDC® COMPUTER SYSTEMS:
CYBER 170 SERIES
MODELS 171, 172, 173, 174, 175
CYBER 70 SERIES
MODELS 71, 72, 73, 74
6000 SERIES

REVISION RECORD

REVISION	DESCRIPTION
A	Manual released.
(11-01-77)	
B	Deletion of CYBER Record Manager documentation on pages 3-1, 3-2, and B-2 through B-7.
(6-20-78)	
Publication No. 84000440	

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
Data Services Publications
P. O. Box 0 HQC02C
Minneapolis, MN 55440

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Front Cover	-								
Title Page/ii	B								
iii/iv	B								
v/vi	A								
vii	A								
1-1 through 1-3	A								
2-1 through 2-21	A								
3-1/3-2	B								
3-3 through 3-5	A								
A-1/A-2	A								
B-1	A								
B-2 through B-7	B								
C-1	A								
Comment Sheet/ Mailer	B								
Back Cover	-								

PREFACE

The Usage Accounting Utility was developed by Control Data Corporation to provide a mechanism by which to bill application products users on a usage basis, rather than by a flat, monthly rental charge. It is intended for use in the user's environment on any of the following systems running under the NOS/BE 1.0 or NOS 1.0 operating system.

- CONTROL DATA® 6000 Computer System;
- CONTROL DATA® CYBER 70 Computer System, Models 71, 72, 73, 74; and
- CONTROL DATA® CYBER 170 Computer System, Models 171, 172, 173, 174, and 175.

Other Control Data publications which may prove helpful to the user include:

Title	Publication Number
NOS/BE Reference Manual	60493800
NOS 1.0 Reference Manual, Volume 1	60435400
NOS 1.0 Reference Manual, Volume 2	60445300
FORTRAN Extended 4 Reference Manual	60497800
COMPASS 3 Reference Manual	60492600
COBOL 4 Reference Manual	60384100
UPDATE Reference Manual	60449400
CYBER Record Manager Version 1 Reference Manual	60495700
Application Installation Handbook	76071100

NOTE

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undefined parameters.

CONTENTS

APPENDIX

FIGURES

TABLES

The Usage Accounting Utility is a two-part package consisting of a usage data reduction and reporting utility program (RANDR) and a set of application interface subroutines (ACOUNTX) which records applications usage.

The RANDR program is a multipurpose, control card callable program which performs the following functions:

- Maintains the product file data base,
- Generates usage detail reports,
- Generates a summary usage report of all product usage since the last billing run,
- Generates a billing report of all product usage since the last billing run, and
- Generates system status reports.

Because RANDR can be employed for user application programs as well as Control Data application products, reports are generated by individual vendor codes with user reports separate from Control Data product reports.

ACOUNTX is a set of subroutines specifically created to provide the interface between the application, the product usage detail file, and the other usage recording media. ACOUNTX is callable only from programs (that is, it is not control card callable) meeting the interface requirements specified in this document. Its main function is to record data concerning the applications usage on the detail file storage medium and log that usage on the user and system dayfiles.

STEPS IN USAGE ACCOUNTING PROCESSING

RANDR PROGRAM PROCESSING

The processing steps involved in the RANDR program depend upon the particular type of run: whether it is an update, summary, or billing run.

An update run updates the product name file by inserting or deleting product name entries, name/address information and/or various usage protection parameters, depending upon the input cards to RANDR.

Billing and summary runs are identical excluding two exceptions: the detail record file is not purged and rebuilt for a summary run as it is for a billing run, and the reporting period and data generated are different for each run.

Billing runs may be made on any day of a given month and cover the period from the first through the last day of the preceding month. Thus, a billing run made on 20 January covers the period of 1 December through 31 December. Subsequent billing runs generate data from the end of the period covered by the previous billing run through the final day of the month to be reported. For example, if the last billing run was made on 20 January (which covered the period of 1 December through 31 December), a billing run on 8 February would report usage from 1 January through 31 January.

The reporting period of summary runs differs from that of billing runs in that usage is reported from the end of the period covered by the previous billing period through the current date and time. In contrast to the preceding billing example, if the previous billing run was on 20 January, a summary run on 8 February would report usage from 1 January through the current time on 8 February.

A billing run made on a date less than one full month past the last billing can be used as an interim billing in order to reduce the usage detail file size on mass storage. It would cover from the end of the previous billing run to the current date and time. For example, if the last billing run was made on 20 January (which covered the period of 1 December through 31 December), a billing run on 27 January would report usage from 1 January through the current time on 27 January. A subsequent run on 8 February would report usage for 27 January through 31 January, but would show the invoice amount for 1 January through 31 January.

Reports from billing runs for Control Data usage priced products must, in some cases, be retained for possible request by Control Data and, in other cases, be forwarded to the Control Data office specified on the reports. Usage detail information and interim summary usage information must be retained as supporting information to the monthly summary usage report which is sent to Control Data. Appropriate instructions appear on all such reports. Reports from summary runs are informational and for the user's utilization.

Determination of the type of run is made by the data deck input to the system. Data deck input may define an update, summary, or billing run. Figures 1-1 and 1-2 illustrate the steps in RANDR processing.

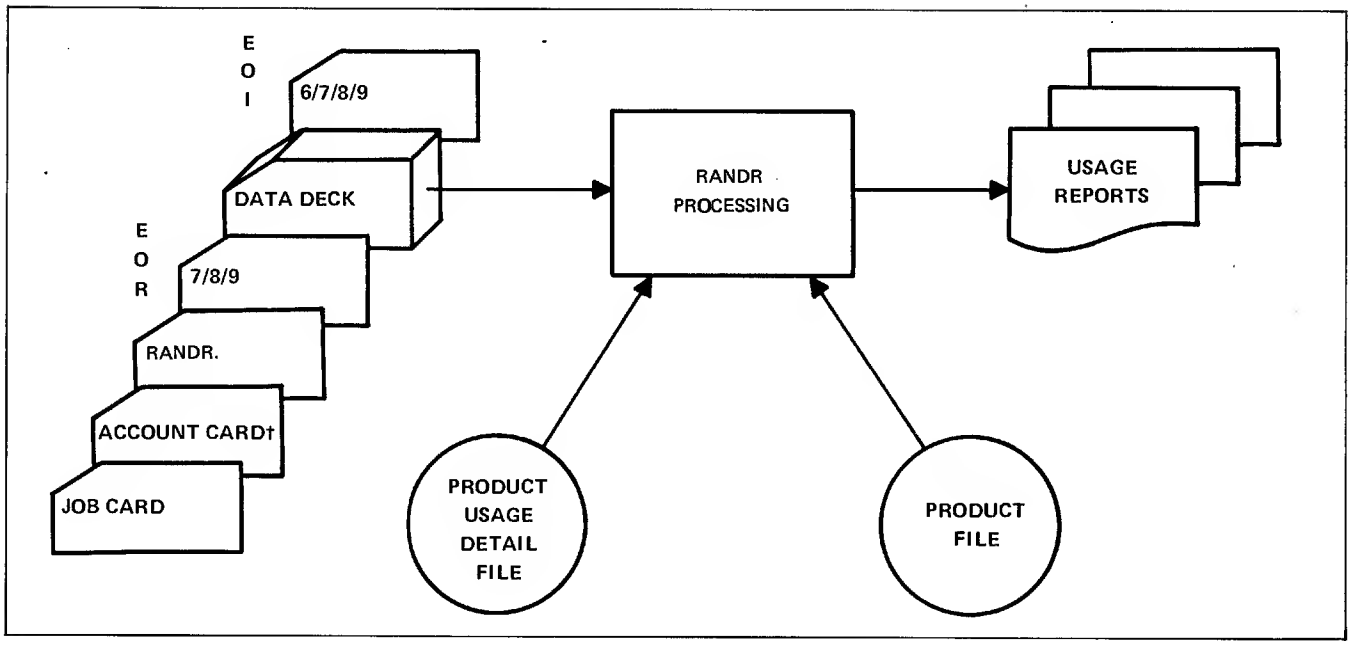


Figure 1-1. RANDR General Processing Chart for Summary or Billing Runs

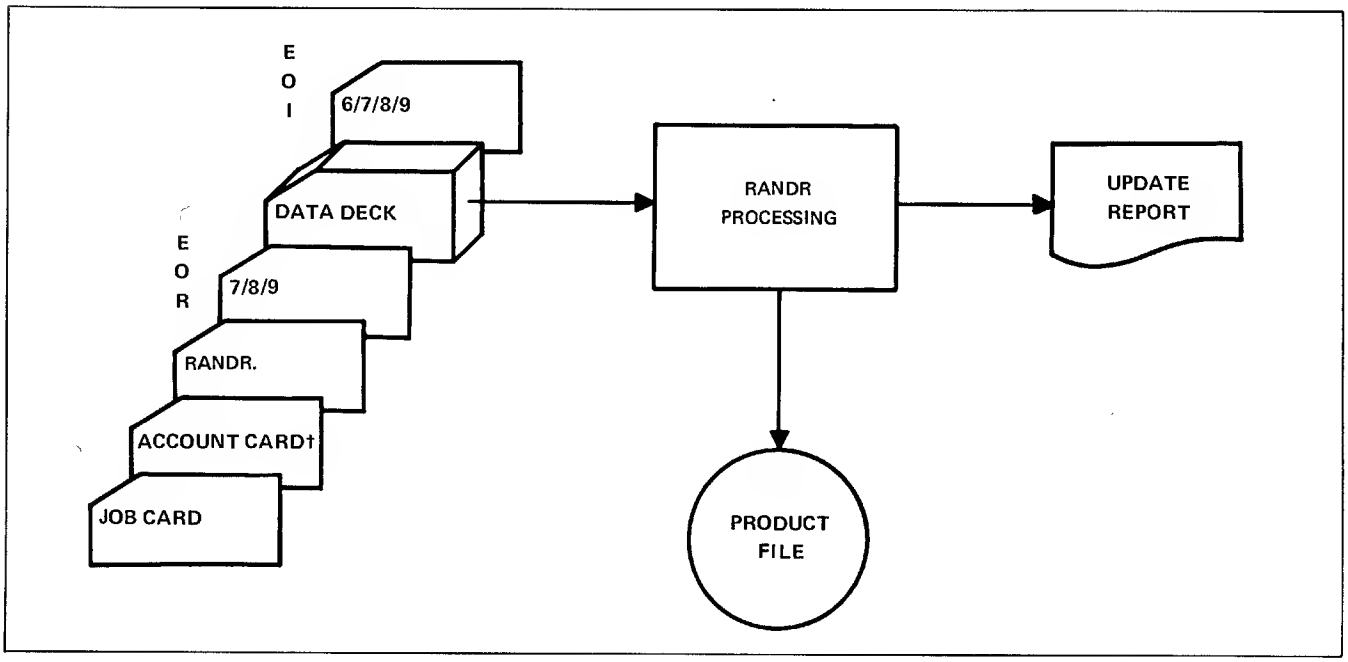


Figure 1-2. RANDR General Processing Chart for an Update Run

†Under NOS operation, whenever the RANDR card is used, the ACCOUNT card must also be used with usernum=ACXLIB. The format is: ACCOUNT, ACXLIB, passwrd, family.

ACOUNTX PROCESSING

Application products leased by Control Data Corporation on a usage basis interface with the usage accumulator portion of the Usage Accounting Utility (ACOUNTX). The primary function of ACOUNTX is to record the actual usage of leased applications. ACOUNTX is stimulated by calls to it by the application with a parameter string defining the operation to be performed and a number of values to be used in logging the transaction. Figure 1-3 shows the general flow of ACOUNTX.

ACOUNTX is program callable from COMPASS, COBOL or FORTRAN Extended programs employing the following call-by-name convention:

Name	Meaning
SA1	Address of the argument list.
+RJ	Subprogram name.
-VFD	12/line number, 18/trace word address

where:

line number = source line number of the statement containing the reference, and

trace word address = address of trace word for the calling routine.

The argument list consists of consecutive words of the form:

Name	Meaning
VFD	60/address of argument
.	.
.	.
.	.
BSSZ	1

COBOL 4, FORTRAN Extended 4 (FTN 4) and COMPASS meet the following calling sequences.

The following form applies to calling ACOUNTX under COBOL:

```
ENTER FORTRAN-X ACOUNTX USING P1, P2, P3, P4, P5,
P6, P7, P8
```

The following form applies to calling ACOUNTX under FTN:

```
CALL ACOUNTX (P1, P2, P3, P4, P5, P6, P7, P8)
```

The following form applies to calling ACOUNTX under COMPASS:

```
TRWD     VFD   42/5HENTER, 18/ENTER
  .       .          .
  .       .          .
  .       .          .
ENTER    EQ    *+1S17        ENTRY POINT
  .       .          .
  .       .          .
  .       .          .
         SA1   PARMADR
+        RJ    ACOUNTX       CALL ACCOUNTING
-        VFD   12/*,18/TRWD
  .       .          .
  .       .          .
  .       .          .
PARMADR  VFD   60/P1          ADDRESS OF FUNCTION
         VFD   60/P2          ADDRESS OF PARAMETER 2
         VFD   60/P3          ADDRESS OF PARAMETER 3
         VFD   60/P4          ADDRESS OF PARAMETER 4
         VFD   60/P5          ADDRESS OF PARAMETER 5
         VFD   60/P6          ADDRESS OF PARAMETER 6
         VFD   60/P7          ADDRESS OF PARAMETER 7
         VFD   60/P8          ADDRESS OF PARAMETER 8
         BSSZ  1              ZERO TERMINATOR
P1       VFD   60/5LSTAUU     START FUNCTION
P2       VFD   60/10H SWNAME  SOFTWARE NAME
P3       VFD   60/4LSWCD      SOFTWARE CODE
P4       CON   250            RATE
P5       CON   0              SURCHARGE
P6       VFD   60/4H          ITEM CODE DESIGNATOR
P7       CON   0              AUU LIMIT
P8       VFD   60/2LCD        VENDOR CODE
```

Section 3 contains a complete description of ACOUNTX functions and parameters.

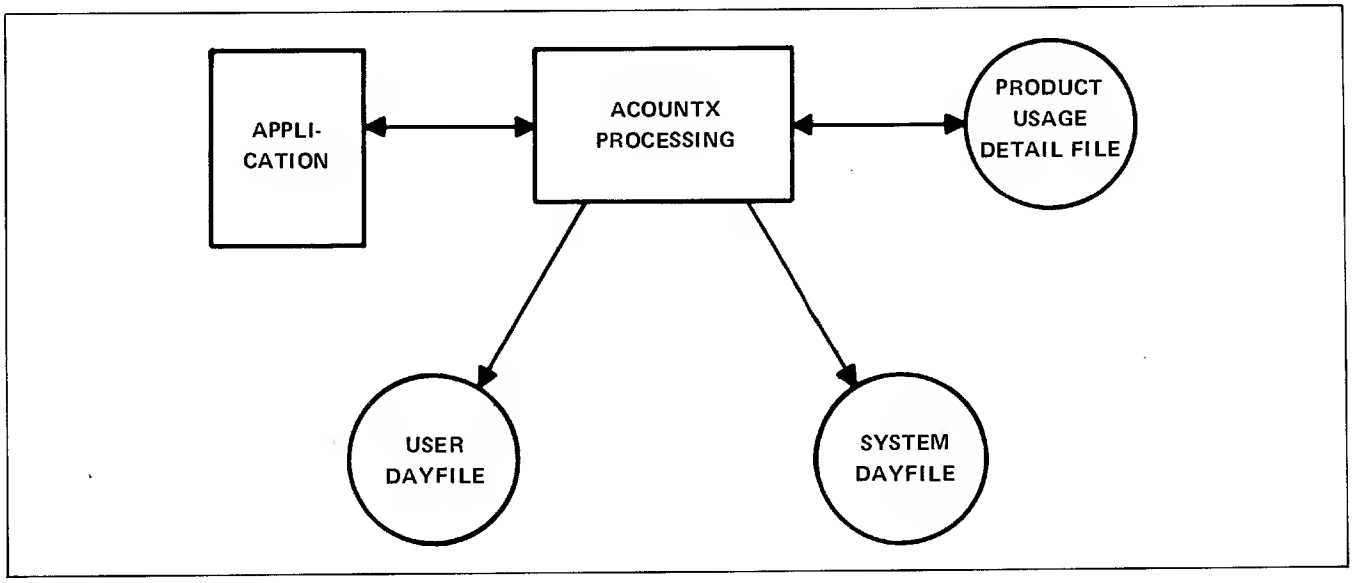


Figure 1-3. ACOUNTX General Process Chart

The usage data reduction and reporting utility program (RANDR) is a bi-functional program within the Usage Accounting Utility. It exists to maintain the product file and produce the usage reports. The specific function to be performed is determined by the data input to RANDR. Figure 2-1 shows a sample deck structure used to call RANDR. The program operates on two files: the usage detail file and the product file. The usage detail file contains information relating to all runs that are usage priced, while the product file contains back-up information regarding vendor, user, and billing period totals for each application product.

TYPES OF RUNS

RANDR processes three types of user requests: update, billing, or summary as specified on the parameter cards in the input data deck (see Data Deck subsection in section 2.)

UPDATE RUNS

Update runs create the product file and perform changes on the entries once the file has been created. Update runs for the purpose of creation are necessary only at the time of initial installation unless the file is lost through destruction of the permanent file system. Whenever it becomes necessary to create the product file, operator permission is requested prior to file initialization by the following message:

```
PAUSE          PRODUCT FILE
LOAD PRODUCT FILE - GO OR DROP
```

An operator response of GO causes a new product name file to be created. If the operator enters DROP the job is dropped.

NOTE

If the product file exists on a dump tape, the file should be restored by a load from the most recent dump tape rather than by an update creation run.

In addition to creating and maintaining the product file, an update run is the only means of establishing the usage detail file. The update phase of RANDR checks for the existence of the usage detail file. If the file is present the update continues; if, however, the file is nonexistent, the following message is displayed on the console:

```
PAUSE          NO DETAIL FILE
LOAD DETAIL FILE - GO OR DROP
```

An operator response of GO causes the usage detail file to be initialized. A DROP response from the operator causes the job to be dropped.

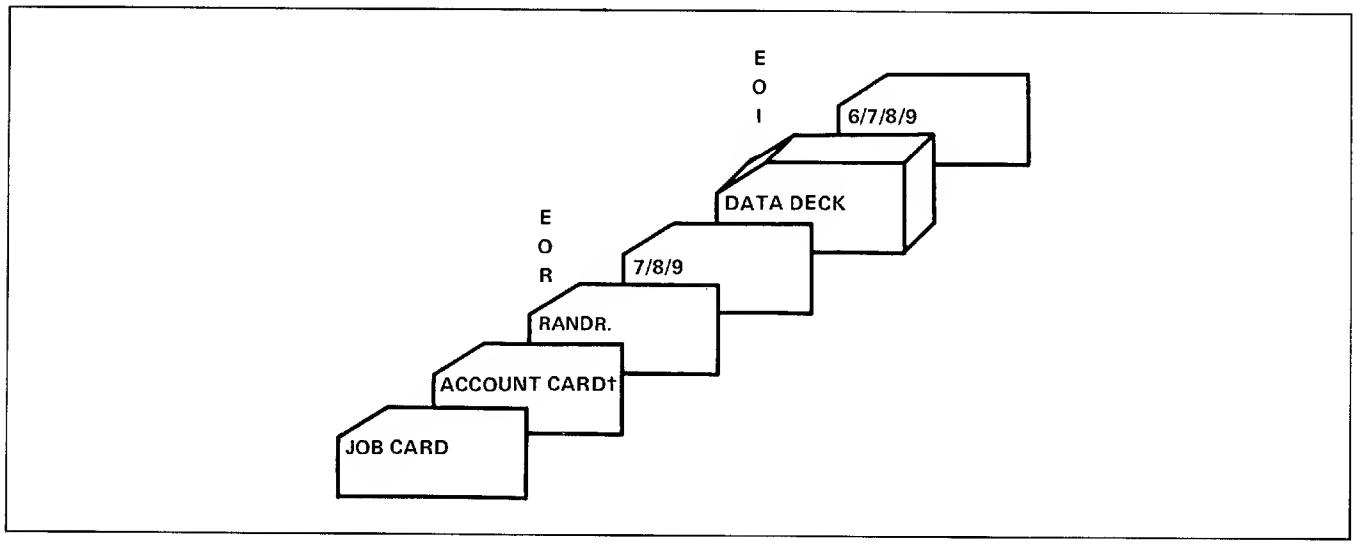


Figure 2-1. Sample Deck to Call RANDR

†Under NOS operation whenever the RANDR card is used the ACCOUNT card must also be used with usernum=ACXLIB. The format is: ACCOUNT, ACXLIB, passwrd, family.

NOTE

If the usage detail file exists on a dump tape, the file should be restored by a load from the most recent dump tape rather than by an update creation run.

After the initial creation of the product file, update runs are used to modify existing information within the file. This is accomplished by the information on the input data deck. User supplied information input to RANDR update via the input data deck consists of the following:

- Number of copies of reports to be generated;
- Control Data Corporation name, address, and contact;
- User name, address, and contact;
- Vendor codes, software codes, and product names for each product; and
- Parameters for usage and reporting protection.

Input transactions are sorted by vendor, software code, product, and card sequence number. The update report shows the processing result of every transaction. Information and run termination messages are shown when necessary.

Because of the importance of the integrity of the product file, operator approval is requested prior to proceeding with any update run. The following message is displayed on the console:

```
PAUSE          UPDATE RUN - GO OR DROP
```

An operator response of GO causes the update run to proceed; an operator response of DROP causes the job to be dropped.

Update runs are made initially to create the product file and detail file; subsequently to add or delete a product, to make a product active or inactive, and/or to alter protection parameters.

Protection parameters include blanking of the account/user number field on reports, operator warning when the detail file has exceeded a specified number of words, and threshold values per product with an overriding control parameter (see Parameter Card subsection in section 2).

Upon termination of an update run an update report is generated indicating the changes made to the product file as well as any errors that may have occurred on the input data deck. A system status report is generated indicating the type of run, its termination status, and the status of the product file and the detail file. (For a description of the various reports, see the Report subsection in section 2.)

BILLING RUNS

Billing runs extract the detail usage records from the usage detail file, update the six-month AUU totals and the accumulated AUU totals for each product in the product file, and produce the usage detail report, the usage summary report (billing report), and the system status report. A billing run is indicated by specifying BILLING on the parameter card (see the Parameter Card subsection in section 2).

Billing runs also perform a clean-up function on the usage detail file by eliminating already totaled detail records. Detail information representing activity subsequent to the billing period is not extracted.

Billing runs can be made at anytime, but a full month constitutes a billing period. Billing runs made at less than full-month intervals also produce a usage detail report, a usage summary report (interim), and a system status report.

In all billing run cases, the detail reports for Control Data products are to be retained. Usage summary reports are, depending upon the elapsed time since the last billing run, to be forwarded to Control Data. Appropriate instructions are generated for each such report. System status reports, and other reports without specific instructions, are informational.

SUMMARY RUNS

Summary runs extract and accumulate usage totals for all products on the product file and produce a summary report of product usage for the past six months up to the current time. Like a billing run, a summary run extracts usage detail records from the usage detail file and accumulates usage liability, but the summary report reflects all usage up to the current date and time. A summary run does not purge any records from the usage detail file like the billing run does. Thus, a summary run will not cause consolidation of the usage detail file to make more rotating mass storage space available. A summary run is specified by the word SUMMARY on the parameter card (see the Parameter Card subsection in section 2.)

In addition to producing summary reports for each vendor code, a system status report is also produced indicating run type and status, usage detail file status, and product file status.

DATA DECK

The data deck is the user's means of communicating directives to RANDR. Through the use of the data cards, the user can specify the type of run (update, summary, or billing), update the customer or CDC mailing address, or update the product file itself. Four card types are accepted by RANDR as input:

- Parameter,
- Control Data record,
- Customer record, and
- Product activity.

RANDR sorts the input by card identifier prior to the start of the processing. Thus the data cards may be in any order.

PARAMETER CARD

The parameter card inputs to RANDR the type of run to be performed and the number of copies to be generated for the various reports.

Format

```
code  ptp  detsz  actp  id  n  type
```

Example

```
CY73  Y  999  N  1  6  UPDATE
```

Column	Parameter	Description
1 to 4	code	A four-character alphanumeric code which identifies the hardware type (CPU model code). This code is used only during the creation of the product and detail files. (The appropriate code is to be chosen from the table in Appendix C.)
5	blank	Reserved.
6	ptp	Product threshold protection indicator, where: Y - specifies that special checking at the initiation and completion of all products is to be done. N - specifies no usage protection or special checking is to be done. Blank - on creation defaults to N; on subsequent updates no change.
7 to 9	blank	Reserved.
10 to 14	detsz	Value for detsz is multiplied by 1000 and specifies the size at which the operator will be notified. Zero implies no checking or warning (see table 2-3). A blank on creation implies 0; on subsequent update runs it implies no change.
15	blank	Reserved.
16	actp	Account/user number protection, where: Y - specifies that the account/user number field will be blanked; N - specifies that the account/user number field will appear on detail reports; Blank - on creation defaults to N; on subsequent update runs indicates no change.
17	blank	Reserved.
18	id	Value for id must be 1. This identifies the card as a parameter card.
19	n	Value for n is 1 to 9. This entry indicates the number of copies of the reports the user desires to receive. Default value is 1.
20 to 26	type	This identifies the type of run to be performed; either UPDATE, BILLING, or SUMMARY, where: UPDATE - allows the above installation information to be initiated and subsequently altered; BILLING - produces a summary report of the previous six-months liability (columns 1 to 17 are ignored). SUMMARY- produces a current product usage liability report (columns 1 to 17 are ignored).
27 to 80	blank	Reserved.

NOTE

On billing runs, the number of report copies produced is the number requested by the user on the parameter card, plus one additional copy for Control Data. For example, if the user specifies a value of 2 for the n parameter on the parameter card, then the run generates three copies of all the reports for each vendor code. The user, in turn, must then forward one copy of the summary report to the Control Data office specified on the reports.

During a RANDR run, the input data deck may consist only of a parameter card, specifying the run type in columns 20 to 26. A single parameter card specifying UPDATE modifies the protection parameters, providing the product file is already established.

For the initial update run, the input consists of the parameter card (id=1), the appropriate number of Control Data and customer record cards (id=2,3), and at least one product card (id=4). Subsequent update runs may consist of any type (id=1,2,3, or 4) of cards.

Errors detected upon input of the parameter card are indicated on the update report (see the Update Report subsection in section 2).

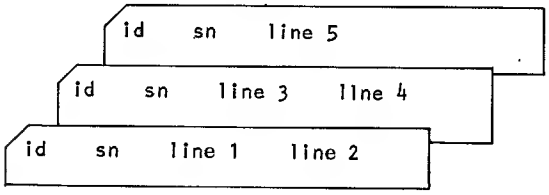
CONTROL DATA RECORD CARDS

The Control Data record cards are a series of three cards which contain the name, address, and contact of the Control Data office to which billing reports are to be sent.

The Control Data cards contain the address of the Control Data regional accounting office to which the summary report is to be sent by the user on a monthly basis. Only one set of Control Data cards may be input to the product name file and it need only be done once. Update errors are displayed on the update report (see the Update Report subsection in section 2).

Errors on input are defined in tables 2-1 and 2-2.

Format



Example

```
2 3 (612)853-4225
2 2 8100 34 AV S. MINNEAPOLIS, MN 55420
2 1 CONTROL DATA CORPORATION MIDWEST REGIONAL SALES OFFICE
```

Column	Parameter	Description
1 to 17	blank	Reserved.
18	id	Value for id must be 2. Identifies this card as a vendor record card.
19	sn	sn is the sequence number of the vendor record card. The values of 1, 2, and 3 are assigned to vendor information as follows: 1 = lines 1 and 2; 2 = lines 3 and 4; and 3 = line 5.

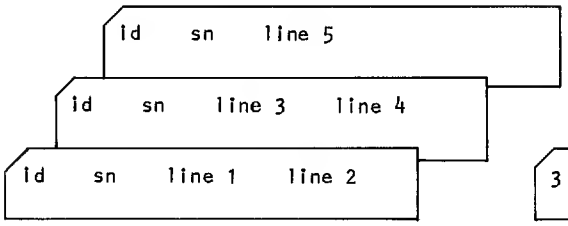
20 to 79	line n	n is a number 1 to 5 representing one of the following five lines of vendor information:
		line 1 = corporation name (columns 20 to 49);
		line 2 = regional accounting office (columns 50 to 79);
		line 3 = street address;
		line 4 = city, state, and zip code; and
		line 5 = area code and phone number.

80	blank	Reserved.

CUSTOMER RECORD CARDS

The customer record cards are a series of three cards containing the name, address, and contact of the user. This information is used by Control Data to identify the user submitting the billing reports.

Format



Example

```
3 3 C/O CRAIG HANSON EXT-3200
3 2 BURLINGTON, MISSISSIPPI 30526 (304)222-6800
3 1 ABACUS DATA CORPORATION 1512 QUINCY AVENUE
```

Column	Parameter	Description
1 to 17	blank	Reserved.
18	id	Value for id must be 3. Identifies this card as a customer record card.
19	sn	sn is the sequence number of the customer record card. The values of 1, 2, and 3 are assigned to customer information as follows: 1 = lines 1 and 2; 2 = lines 3 and 4; and 3 = line 5.

20 to 79	line n	n is a number from 1 to 5 representing one of the following five lines of customer information:
		line 1 = customer name (columns 20 to 49);
		line 2 = street address (columns 50 to 79);
		line 3 = city, state, and zip code (columns 20 to 49);
		line 4 = area code and phone number (columns 50 to 79); and
		line 5 = contact's name and phone extension (columns 20 to 49).

80	blank	Reserved.

Only one set of customer record cards may be input to the product name file. Subsequent runs (with id=3 cards) alter the existing records. Errors detected on input of the customer record cards are displayed on the update report (see the Update Report subsection in section 2).

PRODUCT ACTIVITY CARDS

Product activity cards are the user's means of indicating to RANDR that new products are to be installed on the system (add function); that old products are to be removed from the system (delete function); that a product is to be active or inactive; and/or that the threshold values for usage protection are to be changed.

The information on a product activity card consists of all the information necessary for RANDR to associate a software and vendor code combination, generated by the application, with the product itself.

Format

```
vc   product   id   tr   swc
```

Example

```
VC   COBOL   4   A   ABCD
```

Column	Parameter	Description
1 to 2	vc	A two-character vendor code designating the vendor of the product. The vendor codes CD, OO or ⍽⍽ (⍽=space) are reserved for Control Data's use only. All products with any of these three vendor codes are subject to usage pricing liability.
3 to 12	product	Up to 10 characters designating the product name. Duplication of the product name is permitted by RANDR, but no two products may have the same product name vendor code and software code as well. In the event of duplication, RANDR will merely add the new information to the existing product, thereby dissolving the uniqueness of the two products.
13	blank	Reserved.
14	tr 1	A transaction code used to maintain product file, where: A = add product information to product file; D = delete product information from product file; and Blank = on creation run defaults to A; on subsequent runs this indicates no change.
15	blank	Reserved.
16	tr 2	A transaction code used to alter product status where: A = active (product is active and can be used); I = inactive (product is inactive and no usage will be allowed until status is reset); Blank = on creation, run defaults to A; on subsequent runs, this indicates no change.
17	blank	Reserved.
18	id	Value for id must be 4. This identifies the card as a product activity card.
19	blank	Reserved.
20 to 23	thldd	Value for thldd is multiplied by 1000 and is used as a threshold of usage. When the usage units for a product and a specific software code have accumulated beyond this threshold, the product and its software code are set inactive. No further use is allowed until a noninterim billing run is made, or RANDR is run with appropriate input to reset the product. A warning of this action is put out to the user dayfile, the system dayfile, and the operator (see table 2-3). A 0 is treated as no such protection. A blank implies a 0.

Column	Parameter	Description
24	blank	Reserved.
25 to 29	thldw	Value for thldw is multiplied by 1000 and is used as a threshold of usage. When the usage units for a product and a specific software code have accumulated beyond this threshold, a warning is put out to the user dayfile, the system dayfile, and the operator (see table 2-3). This value must be less than thldd or it is flagged in error. A 0 is treated as no protection. A blank implies a 0.
30 to 80	swc	Software codes for the product. Up to four-character software codes designating a particular job step of the product. Software codes are delimited by commas (,). Software codes of less than four characters are left-justified, blank-filled, by RANDR. [For example, valid software codes are: A, AB, ⍽AB2, A2B3. Invalid software codes are: ABCDEFG (no comma delimiter), ⍽⍽⍽⍽ (reserved for CDC only).]

Product records which have been set inactive as a result of the specified threshold value can be reset with a product activity card specifying I for transaction code 2, and either increasing the threshold value or setting it at 0.

NOTE

If threshold protection was set N via a parameter UPDATE card (id=1), then no usage accumulation checking nor product status checking is performed.

Continuation cards on the product activity cards are not permitted, nor are they needed because two or more product activity cards with the same vendor code and product name are permissible, as long as no duplication of software codes exists. Duplication of vendor and software codes in the input deck causes the second occurrence to be flagged as an error on the update report. That is, two operations on the same record within the same update run are illegal. An update of a record that already exists on the file (that is, all status and protection fields are identical) will still cause a replacement of that record with an indication that the record exists already.

Deletion of records does not take place at the time of the RANDR run, but rather, a status subject to deletion is placed on the record. The record will be deleted after the next billing run is made if all product activity has ceased (see the Usage Summary Report subsection in section 2).

NOTE

Users who for tracking purposes wish to install their own applications under RANDR are cautioned to ensure that no two vendor code/software code combinations are identical. To prevent conflict of vendor products with Control Data leased products, all Control Data products will use one of the following vendor codes: CD, OO or ⍽⍽ (⍽=space). User employment of these codes will cause the usage of the application to be placed on the CDC portion of the reports.

Product activity card information for all Control Data applications subject to usage pricing is supplied to the user by Control Data via the Applications Installation Handbook.

REPORTS

The reports produced by RANDR inform the customer of the status of maintenance runs as well as report usage of all applications under the Usage Accounting Utility. Four reports are produced by RANDR depending upon type of run:

- Update report (update);
- Usage detail report (summary, billing);
- Usage summary report (summary, billing); and
- System status report (update, summary, billing).

A standard header appears on every page of each report. The heading format is shown on each sample report included in this section.

UPDATE REPORT

The update report is an account of all the transactions of an update run. The report is broken up into four parts to coincide with the data card input. Figure 2-2 shows a sample of the update report. The numbers within a circle correspond with the fields described as follows.

Item	Field	Description
1	TITLE	Type of report.
2	PREPARED	Date of preparation.
3	PAGE	Page number.
4	PARAMETER CARD	Indicates that the items following were the input on the parameter card.
5	COPIES	Indicates the number of copies of each report that will be produced upon run termination. NOTE One extra copy of each billing report that is to be sent to Control Data is automatically produced.
6	RUN TYPE	Indicates the type of run request: update, billing, or summary.
7	CARD	The physical position of the card which caused the transaction within the data deck.
8	MESSAGES	Indicates that a questionable situation has been encountered by processing the input card. Table 2-1 contains a list of the error messages which can appear on an update report.
9	CDC RECORD	Name/address fields of Control Data record input. Five lines of information are permissible.
10	CRD SEQ	Sequence number of the vendor or customer record cards.
11	TRANSACTIONS	Update run input directive: add (A) or delete (D), active (A) or inactive (I).
12	ACTION	Action taken by RANDR as a result of the function and directive input. Action taken may be ADDED, DELETED, or IGNORED.
13	CUSTOMER RECORD	Name/address fields of customer record input.
14	PRODUCT ACTIVITY	Indicates that the following section contains all the input transactions against a specific vendor code. The number of sections to the product activity report is dependent upon the number of different vendor codes input during the update run.
15	VENDOR CODE	Two-character mnemonic indicating the product's vendor. Codes CD, 00 and ØØ, are reserved for Control Data products.
16	PRODUCT NAME	Ten-character name associated with the product.
17	SOFTWARE CODE	One- to four-character mnemonic assigned to a software product job step. One product may have any number of software codes assigned to it. The number is determined by the structure of the application program itself.
18	UNIDENTIFIABLE CARDS	When a card is input that cannot be fully processed during the update run, the card image of the erroneous card is displayed in this section.

(1) U P D A T E R E P O R T

(2) PREPARED - 77/09/18 A P P L I C A T I O N U S A G E A C C O U N T I N G (3) P A G E 1

VENDOR - 99 CUSTOMER

(4) P A R A M E T E R C A R D

PROTECTION PARAMETERS	(5) COPIES	(6) RUN TYPE	(7) CARD	(8) MESSAGES
THLD- Y UOF-00010000 ACCTG-N	1	UPDATE	1	

(9) C D C R E C O R D

NAME/ADDRESS	(10) CRD SEQ	(11) TRANSACTION	(12) ACTION	CARD	MESSAGES
CONTROL DATA CORPORATION	1	ADD	ADDED	2	
MIDWEST REGIONAL SALES OFFICE	1	ADD	ADDED	2	
8100 34TH AVE S	2	ADD	ADDED	3	
MINNEAPOLIS, MN 55420	2	ADD	ADDED	3	
PHONE - 612/853-3800	3	ADD	ADDED	4	

(13) C U S T O M E R R E C O R D

NAME/ADDRESS	CRD SEQ	TRANSACTION	ACTION	CARD	MESSAGES
ABACUS CORPORATION	1	ADD	ADDED	5	
1512 QUINCY AVENUE	1	ADD	ADDED	5	
BURLINGTON, MISSISSIPPI 30526	2	ADD	ADDED	6	
PHONE - 304/222-6800	2	ADD	ADDED	6	
C/O CRAIG HANSON EXT - 3200	3	ADD	ADDED	7	

(14) P R O D U C T A C T I V I T Y

(15) VENDOR CODE	(16) PRODUCT NAME	(17) SOFTWARE CODE	TRANSACTION	ACTION	CARD	THRESHOLD	MESSAGES
CD	APEX III	APEX	ADD -A	ADDED	8	7/ 4	
CD	APEX III	APE4	ADD -A	ADDED	8	7/ 4	
CD	APEX III	APE5	ADD -A	ADDED	8	7/ 4	
CD	APEX III	APE7	ADD -A	ADDED	8	7/ 4	
CD	TEST-PROD	BILL	ADD -A	ADDED	18	10/ 5	
CD	GARBAGE	FTGA	-I	IGNORED	14	30/ 10	01 - SWCE NOT FOLLOWED BY COMMA
CD	GARBAGE	R	ADD -A	ADDED	14	30/ 10	
CD	STAUU	ST01	ADD -A	ADDED	13	300/ 275	
CD	STAUU	ST02	ADD -A	ADDED	13	300/ 275	

Figure 2-2. Update Report

UPDATE REPORT

PREPARED - 77/09/18 APPLICATION USAGE ACCOUNTING PAGE 2

VENDOR - 99 CUSTOMER

(14) PRODUCT ACTIVITY

VENDOR CODE	PRODUCT NAME	SOFTWARE CODE	(11) TRANSACTION	(12) ACTION	(7) CARD	THRESHOLD	(8) MESSAGES
(15) CD	(16) STAUU	(17) ST03	ADD -A	ADDED	13	300/ 275	
CD	STAUU	ST08	ADD -A	ADDED	13	300/ 275	
CD	TEST-PROD	SWC1	ADD -A	ADDED	18	10/ 5	
CD	TEST-PROD	SWC2	ADD -A	ADDED	18	10/ 5	
CD	TEST-PROD	SWC4	ADD -A	ADDED	18	10/ 5	
CD	TEST-PROD	SWC5	ADD -A	ADDED	18	10/ 5	
CD	TEST-PROD	Y	ADD -A	ADDED	18	10/ 5	
CD	TEST-PROD	YYYY	-I	IGNORED	18	10/ 5	01 - SWCE NOT FOLLOWED BY COMMA
CD	TEST-PROD	YYYY	-I	IGNORED	18	10/ 5	01 - SWCE NOT FOLLOWED BY COMMA
CD	TEST-PROD	ZZZZ	ADD -A	ADDED	18	10/ 5	
CD	TEST-PROD	ZZZZ	ADD -A	IGNORED	18	10/ 5	12 - DUPLICATE PRODUCT/SWCE CODES

PRODUCT ACTIVITY

VENDOR CODE	PRODUCT NAME	SOFTWARE CODE	TRANSACTION	ACTION	CARD	THRESHOLD	MESSAGES
CP	TESTCOM	CMP1	ADD -A	ADDED	11	200/ 100	
CP	TESTCOM	CMP2	ADD -A	ADDED	11	200/ 100	

PRODUCT ACTIVITY

VENDOR CODE	PRODUCT NAME	SOFTWARE CODE	TRANSACTION	ACTION	CARD	THRESHOLD	MESSAGES
FT	FORTRAN	FTN1	ADD -I	ADDED	10	100/ 60	
FT	FORTRAN	FTN2	ADD -I	ADDED	10	100/ 60	

PRODUCT ACTIVITY

VENDOR CODE	PRODUCT NAME	SOFTWARE CODE	TRANSACTION	ACTION	CARD	THRESHOLD	MESSAGES
UU	SAMPLE	SAMP	ADD -A	IGNORED	12	50/ 60	17 - THRESHOLD VALUES INCONSISTENT
UU	SAMPLE	SAMP	ADD -A	ADDED	15	50/ 30	
UU	SAMPLE	TA01	ADD -A	ADDED	9	90/ 60	

Figure 2-2. Update Report (Cont'd.)

```
                                    U P D A T E   R E P O R T

PREPARED - 77/09/18        A P P L I C A T I O N   U S A G E   A C C O U N T I N G        P A G E    3

VENDOR - 99                                                                    CUSTOMER



P R O D U C T   A C T I V I T Y

VENDOR CODE    PRODUCT NAME    SOFTWARE CODE    TRANSACTION    ACTION    CARD    THRESHOLD          MESSAGES

     VC           COBOL            CB01          ADD  -A       ADDED      16     3333/  20
     VC           COBOL            CB02          ADD  -A       ADDED      16     3333/  20
     VC           COBOL            ST02          ADD  -A       ADDED      16     3333/  20
     VC           COBOL            ST03          ADD  -A       ADDED      16     3333/  20
     VC           COBOL            ST06          ADD  -A       ADDED      16     3333/  20
     VC           COBOL            ST07          ADD  -A       ADDED      16     3333/  20

(18) U N I D E N T I F I A B L E   C A R D S

                                  CARD IMAGE                                        MESSAGES

     CDTEST-PROD      UASWC1,SWC2,SWC4,SWC5,YYYYYYYYY,ZZZZ,ZZZZ,ZZZZ,BILL
```

Figure 2-2. Update Report (Cont'd.)

USAGE DETAIL REPORT

The usage detail report contains a sequential list of all application product job steps subject to usage pricing since the last billing run was made. Entries in the detail report are arranged in a chronological order by software code within product name. Each vendor code produces a separate detail report. Usage liability is itemized by job step, total per software code, total per product, and total per vendor. Figure 2-3 shows a sample usage detail report. This report is put out for both summary and billing runs, with special instructions for billing runs. The period covered by the summary run is from the end of the previous billing period through the current date and time. The period covered by the billing run is from the end of the previous billing period through the end (or current date in case of interim billing) of the billing month.

NOTE

Usage detail reports from billing runs for vendor code CD must be retained.

Item	Field	Description
1	TITLE	Type of report.
2	PREPARED	Date of preparation.
3	PAGE	Page number.
4	VENDOR: CDC ADDRESS	Vendor code of report section; Control Data name and address.
5	SPECIAL INSTRUCTIONS	When present, this indicates that one copy is retained or sent to the Control Data address specified under the vendor code.
6	CUSTOMER	Name and address of the customer.
7	PRODUCT NAME	Alphabetical list of products (one to 10 charachters) per vendor.
8	SOFTWARE CODE	Alphabetical list of a software codes per product (one to four characters).
9	JOB BANNER	Job name of job generating usage.
10	ACCOUNT/USER NUMBER	Account or user number generating usage.
11	RUN DATE	Date of run.
12	START/END TIME	Time of start and end of job step.
13	AUU STEP QUANTITY	Accumulated number of application usage units (AUU).
14	AUU TOTALS	Total accumulated number of application usage units per software code, product, and vendor code. AUU totals are rounded to the nearest integer value; thus the total AUU for a particular software code may be 0.
15	Unlabeled	Run error indicators (first three characters of the field).
16	Unlabeled	Nine-character record checksum.
17	Software code total	Software code AUU total per product.
18	PRODUCT TOTAL	Product AUU total per vendor code.
19	VENDOR TOTAL	The total number of AUU used for all products within the vendor code.

(1) USAGE DETAIL REPORT

(2) PREPARED - 77/10/15 APPLICATION USAGE ACCOUNTING (3) PAGE 1

(4) VENDOR - CO

CONTROL DATA CORPORATION
MIDWEST REGIONAL SALES OFFICE
8100 34TH AVE S
MINNEAPOLIS, MN 55420
PHONE - 612/853-3800

(5) THIS DETAIL REPORT MUST BE RETAINED TO BE MADE AVAILABLE UPON CONTROL DATA REQUEST CODE C175

CUSTOMER

(6) ABACUS CORPORATION
1512 QUINCY AVENUE
BURLINGTON, MISSISSIPPI 30526
PHONE - 304/222-6800
C/O CRAIG HANSON EXT - 3200

(7) PRODUCT NAME	(8) SOFTWARE CODE	(9) JOB BANNER	(10) ACCT/USER NUMBER	(11) RUN DATE	(12) START/END TIME	(13) AUU STEP QUANTITY	(14) AUU TOTALS	(15) (16)
APEX III	APE7	JOBZ33W	0414108	77/09/18	16.05.12/16.07.41	119.718		F0010940 2400
	APE7	JOBZ23Y	0414108	77/09/18	16.08.58/16.10.23	59.081	(17) 179	0009003 01500
	****	APEX III		TOTAL			(18) 179	
NO*PRODUCT	SAMP	ZFTN63V	0414108	77/09/18	16.03.45/16.03.45	10.163	(17) 10	0000000 00200
NO*PRODUCT	TA01	BECB63X	0414108	77/09/18	16.10.35/16.10.35	0.008		0000010 00000
	TA01	BECB63X	0414108	77/09/18	16.10.35/16.10.46	93.930	(17) 94	0007080 01300
	****	NO*PRODUCT		TOTAL			(18) 104	
STAUU	ST01	YFTN63U	0414108	77/09/18	16.02.52/16.02.52	20.195	(17) 20	0000040 00100
STAUU	ST02	YFTN63U	0414108	77/09/18	16.02.52/16.02.52	0.011	(17) 0	0000010 00000
STAUU	ST03	YFTN63U	0414108	77/09/18	16.02.53/16.02.53	20.128	(17) 26	0000300 01000
	****	STAUU		TOTAL			(18) 40	
VENDOR TOTAL - CURRENT BILLING PERIOD							(19) 323	

Figure 2-3. Usage Detail Report

USAGE SUMMARY REPORT

The usage summary report contains the accumulative total of AUUs for each product and software code within the vendor code from the usage detail report. This report is produced for summary and billing runs. The difference between the reports for each type of run is that for billing runs, only records from the detail file with dates prior to the end of the billing month are reflected in the report, whereas for summary runs, all records from the detail file are reflected in the report. In addition, the report shows the number of AUUs used for all software codes over the previous five months and the total year-to-date number of AUUs since the software code was installed on the system.

Figure 2-4 contains a sample usage summary report. Key items are listed below.

Item	Field	Description
1 to 6	Standard Heading	Reports to be sent to Control Data will have item 5 filled in with special instructions, while reports that are for the customer use only will have this field blank.
7	PRODUCT NAME	Alphabetical list of products per vendor (one to 10 characters).
8	SOFTWARE CODE	Alphabetical list of software codes per product (one to four characters).
9	EFFECTIVE DATE	Date that the software code was entered into the product name file.
10	STATUS	Software code special disposition status, where: S - indicates that the user wished the software code to be deleted but it was not due to usage activity during the current billing month. The software code will be dropped to D status during the next billing month for which the software code shows no usage activity. D - indicates that the user wishes the software code to be deleted. If the software code shows no activity during the month, the code will be deleted from the product name file and will not appear on future reports.
11	Y-T-D AUU USAGE	Accumulated AUU Usage - indicates the total number of AUUs consumed since the software code was entered into the product name file or for the current year.
12	PREVIOUS BILLING PERIODS	Total number of AUUs consumed by the software code for the months indicated. The five previous billing periods are displayed.
13	CURRENT	Total number of AUUs consumed during the current month for summary runs or the total for the last full month for billing runs.
14	TOTAL USAGE	Y-T-D and monthly AUU totals of all software codes within a product.
15	*	Total usage entries followed by an asterisk (*) indicate that a billing run was not made during these months and that the customer will be billed this month for the usage.
16	INV. AMT.	Invoice amount - indicates usage liability for current billing period.
17	ACCUM PRODUCT USAGE	Total Y-T-D and monthly usage for vendor code.

(1) U S A G E S U M M A R Y R E P O R T

(2) PREPARED - 77/10/15 A P P L I C A T I O N U S A G E A C C O U N T I N G (3) P A G E 1

VENDOR - CO

(4) CONTROL DATA CORPORATION
MIDWEST REGIONAL SALES OFFICE
8100 34TH AVE S
MINNEAPOLIS, MN 55420
PHONE - 612/853-3800

(5) FORWARD TO THE ATTENTION OF
COC REGIONAL ADMINISTRATION
AT NOTED COC ADDRESS
CODE C175

CUSTOMER

(6) ABACUS CORPORATION
1512 QUINCY AVENUE
BURLINGTON, MISSISSIPPI 30526
PHONE - 304/222-6800
C/O CRAIG HANSON EXT - 3200

(7) PRODUCT NAME	(8) SOFTWARE CODE	(9) EFFECTIVE DATE	(10) STATUS	(11) Y-T-D AUU USAGE	(12) --PREVIOUS BILLING PERIOD AUU- APRIL	MAY	JUNE	JULY	AUGUST	(13) CURRENT BILLING PERIOD AUU SEPTEMBER	
APEX III	APEX	77/09/18		0	0	0	0	0	0	0	
	APE4	77/09/18		0	0	0	0	0	0	0	
	APE5	77/09/18		0	0	0	0	0	0	0	
	APE7	77/09/18		179	0	0	0	0	0	179	(16)
	(14)	TOTAL USAGE		179	0	0	0	0	0 (15)	179	INV. AMT.
GARBAGE	R	77/09/18		0	0	0	0	0	0	0	
		TOTAL USAGE		0	0	0	0	0	0	0	INV. AMT.
NO*PRODUCT	SAMP	00/00/00	S	10	0	0	0	0	0	10	
	TA01	00/00/00	S	94	0	0	0	0	0	94	
		TOTAL USAGE		104	0	0	0	0	0	104	INV. AMT.
STAUU	ST01	77/09/18		20	0	0	0	0	0	20	
	ST02	77/09/18		0	0	0	0	0	0	0	
	ST03	77/09/18		20	0	0	0	0	0	20	
	ST08	77/09/18		0	0	0	0	0	0	0	
		TOTAL USAGE		40	0	0	0	0	0	40	INV. AMT.
TEST-PROD	BILL	77/09/18		0	0	0	0	0	0	0	
	SWC1	77/09/18		0	0	0	0	0	0	0	
	SWC2	77/09/18		0	0	0	0	0	0	0	
	SWC4	77/09/18		0	0	0	0	0	0	0	
	SWC5	77/09/18		0	0	0	0	0	0	0	
	Y	77/09/18		0	0	0	0	0	0	0	
	ZZZZ	77/09/18		0	0	0	0	0	0	0	
		TOTAL USAGE		0	0	0	0	0	0	0	INV. AMT.
(17) ****** ACCUM PRODUCT TOTAL				323	0	0	0	0	0	323	

Figure 2-4. Usage Summary Report From Billing Run

SYSTEM STATUS REPORT

The system status report contains information relating to the type of run and the status of the usage accounting files. This report is produced regardless of the run type. Figure 2-5 contains a sample system status report. Contents of the data fields are:

Item	Field	Description
1 to 6	Standard Heading	This report never contains special instructions in item 5. Therefore, the report need not be sent to Control Data.
7	AUU SYSTEM MONTH	Month for which the run applies. For billing runs the AUU SYSTEM MONTH will be the last full month or part thereof. For summary runs, it includes the current system month as well.
8	LAST BILLING MONTH	The system month of the last billing run.
9	AUU PRODUCT FILE STATUS	The status of the product name file where: NEW = file allocated during run. INACTIVE = file allocated previously but empty. ACTIVE = file allocated and not empty.
10	AUU DETAIL FILE STATUS	The status of the detail file where: NEW = file allocated during run. INACTIVE = file allocated previously but empty. ACTIVE = file allocated and not empty.
11	TYPE OF RUN	Indicates the type of run requested: BILLING, UPDATE or SUMMARY.
12	RUN TERMINATION	The status of the run is shown as NORMAL or ABNORMAL. Runs with NORMAL termination may have had errors upon input, but the errors were not fatal. Abnormal run termination occurs for fatal errors.
13	Unlabeled	For no errors, no message appears. For a trivial or fatal error this item contains a diagnostic message and an error code explaining the error. (A summary of these error codes and diagnostic messages is contained in table 2-2 of the error message section.)

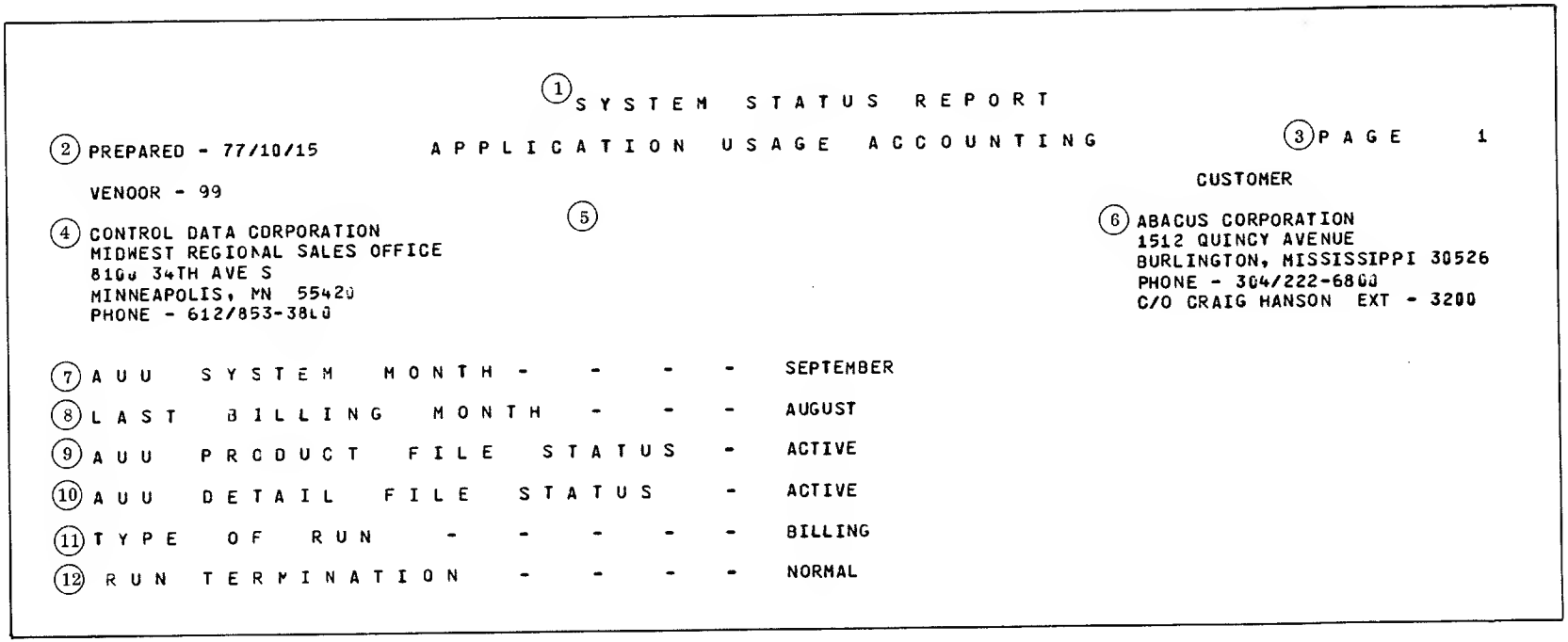


Figure 2-5. System Status Report

ERROR MESSAGES

Error codes and error messages appear on the updatc report and the system status report whenever a questionable situation is encountered during the processing of an input deck transaction. Diagnostic messages may be trivial or fatal. For all trivial errors, RANDR continues processing the remaining transactions; fatal errors cause an abnormal run termination indicator on the update status report.

Table 2-1 lists the errors that appear on the update report as a result of bad input data. Error types may be trivial (T) or fatal (F) as indicated in table 2-1. Trivial errors produce NORMAL run termination. Fatal errors produce ABNORMAL run termination and a fatal error code and message as listed in table 2-2. Fatal errors appear on the system status report.

TABLE 2-1. UPDATE REPORT ERRORS

Error Code/Message	Error Type	Cause	Corrective Action
01-SWC NOT FOLLOWED BY COMMA	T	More than four characters encountered on product activity card without a comma.	Correct software code in error and rerun update.
02-SEQUENCE NUMBER ERROR	F	Sequence number on customer vendor record cards greater than 3.	Correct card and rerun.
03-PRODUCT CODE INVALID	T	Product name on product activity card invalid. Product name of all blanks is invalid.	Correct card and rerun.
04-SOFTWARE CODE INVALID	T	Software code or product activity card invalid. All blanks or all zeros is invalid.	Correct card and rerun.
05-TRANSACTION CODE INVALID	T	Code not a A or D.	Correct code and rerun.
06-RECORD IS NOT ON FILE	T	Attempted to delete software code for a product which is nonexistent.	Possible keypunch error. Correct if necessary.
07-RECORD ALREADY ON FILE	T	Attempted to add a software code/product/vendor code which exists from previous run.	None.
08-PROTECTION PARAMETER ERR	T	Parameter is ignored.	Correct deck and rerun.
09-INVALID RUN PARAMETER	F	Run type not BILLING, SUMMARY, or UPDATE on parameter card.	Correct parameter and rerun.
10-DETAIL SIZE PARAMETER ERR	T	Detail size specified; probably nonnumeric.	Correct deck and rerun.
11-COPY PARAMETER ERR-1 ASSUMED	T	Nonnumeric copies specified on parameter card.	Run will continue and use a default of 1 as number of report copies to produce.
12-DUPLICATE VENDOR/SWC CODES	T	Vendor/software code combination already exists on file.	Run continues.
13-DUPLICATE SEQUENCE NUMBER	F	More than one card of a vendor or customer record has the same sequence number.	Correct vendor or customer record cards and rerun.
14-IDENT CODE INVALID	T	id code on input card greater than 4.	correct invalid card and rerun.

TABLE 2-1. UPDATE REPORT ERRORS (Cont'd)

Error Code/Message	Error Type	Cause	Corrective Action
15-THRESHOLD VALUE NOT NUMERIC	T	Nonnumeric value other than blanks found in threshold values.	Correct invalid card and rerun.
16-REPLACES PREVIOUS CARD	T	Duplicate parameter cards found. Last one overrides.	None.
17-THRESHOLD VALUES INCONSISTENT	T	Value specified for product deactivation threshold not larger than that threshold specified for warning.	Correct invalid card and rerun.

TABLE 2-2. SYSTEM STATUS REPORT ERRORS

Error Code/Message	Cause	Corrective Action
01-Reserved		
02-NO DATA CARDS	No input found for RANDR run.	Correct deck and rerun.
03-REJECT RELEASING FILE	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.
04-REJECT CREATING FILE	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.
05-PARAMETER CARD	Parameter card missing from update input deck.	Correct deck and rerun.
06-MISSING CDC RECORD	One or more Control Data record cards missing during update run.	Insert missing Control Data cards and rerun.
07-NO CDC RECORDS	No Control Data address record cards found.	Insert cards and rerun.
08-CARD SEQUENCE NUMBER ERR.	Sequence numbers on data cards improperly sequenced.	Correct cards and rerun.
09-MISSING CUSTOMER RECORD	Customer record not input or there are not three customer record cards input during update run.	Input complete set of customer record cards.
10-INVALID INPUT CARD DECK	Input card in data deck not in correct format.	Correct card(s) and rerun.
11-DUPLICATE PARAMETER CARD	More than one parameter card in input deck.	Remove duplicate card and rerun.
12-INVALID RUN PARAMETER	Run type parameter is other than UPDATE, BILLING or SUMMARY.	Correct parameter card and rerun.
13-INVALID INPUT DECK	Input deck for RANDR found to be invalid.	Correct deck and rerun.
14-CUSTOMER RECORD ERROR	Customer record card not formatted correctly on update run.	Correct error and rerun.
15-VENDOR RECORD ERROR	Vendor record card not formatted correctly on update run.	Correct error and rerun.

TABLE 2-2. SYSTEM STATUS REPORT ERRORS (Cont'd)

Error Code/Message	Cause	Corrective Action
16-NO PRODUCT FILE	Product file not in system.	Load product file from last dump tape. If no dump tape exists, run update with product activity card and type GO when product file message appears.
17-PRODUCT FILE EMPTY	No data in product file on billing or summary run.	Run update with product activity cards.
18-NO DETAIL FILE	Detail file not in system.	Load detail file from last dump tape. If no dump tape exists, run update and type GO when detail file message appears.
19-DETAIL FILE EMPTY	No data in detail file during a billing or summary run.	Summary or billing run, but no detail records have accumulated yet. This particular situation is not always an ABNORMAL fatal error. If the detail file is in good order (NORMAL termination) but contains no detail usage entries, all reports except the detail usage reports are generated. ABNORMAL termination indicates that the detail file is invalid and probably needs to be recreated.
20-REJECT RETURNING FILE	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.
21-INVALID SYSTEM DATE	Operating system date not in valid range.	Deadstart system and enter correct date.
22-AUU/SYSTEM DATES CONFLICT	Date of last billing run is not at least one month prior to current system month.	Wait until month change occurs to run billing run.
23-EXCESS BILLING PERIODS	A billing run has not been run in last six months, therefore, itemized monthly INV. AMT. are not available for month previous to the sixth previous months.	Billing runs must be run at least every six months to maintain monthly liability.
24-REJECT CATALOGING FILE	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.
25-REJECT ON FILE RENAME	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.
26-TEMP PRODUCT FILE EXISTS	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.
27-TEMP DETAIL FILE EXISTS	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.
28-REJECT PURGING FILE	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.
29-FATAL ERRORS IN UPDATE	Fatal errors found during update run.	Check update report. Correct problem and rerun.
30-INVALID MACHINE TYPE	Operating system interface failure.	See Product and Usage Detail Files description in Application Installation Handbook.

TABLE 2-3. OPERATOR WARNING MESSAGES

Error Code/Message	Cause	Corrective Action
PAUSE NO PRODUCT FILE LOAD PRODUCT FILE - GO OR DROP	The product file was not found for a RANDR run.	Type n. DROP, reload last product file and rerun or type n. GO and continue.
PAUSE NO DETAIL FILE LOAD DETAIL FILE - GO OR DROP	The detail file was not found for a RANDR run.	Type n. DROP, reload last product file and rerun or type n. GO and continue.
PAUSE UPDATE RUN - GO OR DROP	An update run has been specified by the input to RANDR.	Verify a product file update was requested and type n. GO.
VC swc PRODUCT DEACTIVATED	The deactivate threshold has been exceeded and the product deactivated.	Run full month billing run or RANDR to reset status and threshold.
VC swc PRODUCT THRESHOLD REACHED	The warning threshold has been exceeded but the product status remains active.	No action necessary.
USAGE DETAIL FILE SIZE EXCEEDED - TYPE N. GO	The specified number of words has been reached on the detail file size.	Type n. GO; make billing run if disk space is required.

ACOUNTX is the sole interface which a usage priced application, whether that of Control Data or of the user, has with the accounting files and the accounting system. ACOUNTX is available for programs written in FORTRAN Extended 4 (FTN 4), COBOL 4 and COMPASS 3.

Generally, an application will make two calls to ACOUNTX, one to start application usage accounting, and the second to end accounting for each job step within the run. However, intermediate calls to change application usage unit (AUU) rates or limits may also be made between the start/end envelope as need requires. Change rate and change limit functions are also convenient methods for the application itself to obtain the current AUU liability and the number of system seconds (SS) consumed thus far.

A system second, as used in this text, is a unit of measure of central processor activity. It represents the central processor resources used for the run multiplied by a constant. This constant is derived such that the number of system seconds remains the same for a given application run regardless of the particular mainframe the application was executed on.

Applications which end prematurely due to a system abnormal condition will have accounting terminated without the application making the end call. End processing under these circumstances causes end termination with error messages appearing on the system and user dayfiles and an error code placed in the job step associated accounting detail record in the usage accounting detail file. In addition, upon occurrence of an abnormal abort situation, the register file and a dump of 100 octal words before and after the location where the error was detected are dumped to the output file to aid in locating the software/hardware problem (see Abnormal Errors subsection in section 3).

APPLICATION USAGE ACCOUNTING CALLS

Application calls to ACOUNTX follow the standard call-by-name convention specified for COBOL 4, FTN 4 and COMPASS 3. Up to eight parameters are accepted by ACOUNTX; one specifying the function to perform and up to seven other parameters, depending upon the function. The general forms of all calls to ACOUNTX are as follows:

- FORTRAN Extended (FTN 4):

```
CALL ACOUNTX (ICODE, ISFNAME, ISCODE, IRATE, ISCHARG,
   ISITEM, IALIM, IVENCD).
```

- COBOL 4:

```
ENTER FORTRAN-X ACOUNTX USING ICODE, ISFNAME, ISCODE,
   IRATE, ISCHARG, ISITEM, IALIM, IVENCD.
```

- COMPASS:

```
      SA1   PADDR
+     RJ    ACOUNTX
-     VFD   12/Line Number, 18/Trace Word Address
        .     .
        .     .
        .     .
PADDR VFD   60/ICODE     ADDRESS OF FUNCTION
                         PARAMETER
      VFD   60/ISFNAME   ADDRESS OF PRODUCT NAME
                         PARAMETER
      VFD   60/ISCODE    ADDRESS OF SOFTWARE CODE
                         PARAMETER
      VFD   60/IRATE     ADDRESS OF SOFTWARE RATE
                         PARAMETER
      VFD   60/ISCHARG   ADDRESS OF SURCHARGE RATE
                         PARAMETER
      VFD   60/ISITEM    ADDRESS OF ITEM CODE
                         PARAMETER
      VFD   60/IALIM     ADDRESS OF AUU LIMIT
                         PARAMETER
      VFD   60/IVENCD    ADDRESS OF VENDOR CODE
                         PARAMETER
      BSSZ  1
```

PARAMETER DEFINITION

The parameter list specification of calls to ACOUNTX is variable in length with fixed-position parameters. Null fields should not be used in the parameter for FTN 4 and COBOL 4 (that is, (,,) should not be used). Instead (,0,) or (,b,) should be used to indicate the presence of a null parameter in the string (where b indicates a space). This technique may be used for any optional parameter which the application does not need or for a parameter that ACOUNTX does not use for a specific function call.

- ICODE (left justified, blank, or zero fill). A literal that defines the function to perform (5 or 6 characters).

  1. Start AUU accounting.

     5LSTAUU, 5HSTAUU

  2. Change AUU rate. (Used also by the application to obtain the number of AUUs consumed by the application.)

     5LCRAUU, 5HCRAUU

3. Change AUU limit. (Used by the application to obtain the number of system seconds consumed by the application.)

   6LLIMAUU, 6HLIMAUU

4. End AUU accounting.

   5LEDAUU, 5HEDAUU

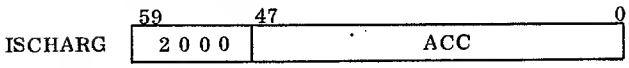
- ISFNAME (left justified, blank, or zero fill). A literal (one to 10 characters) used to designate the name of the software product. This literal can, but need not, correspond to the product name on the product name file. The literal is displayed in the system dayfile and user dayfile in the start messages associated with the application run.

- ISCODE (left justified, blank, or zero fill). A literal used as the software code to designate a particular job step within a vendor product (one to four characters). ISCODE may be duplicated within the system, however, the combination of software code (ISCODE) and vendor code (IVENCD) must be unique. RANDR checks to guarantee that no two vendor code/software code combinations are duplicated.

- IRATE (integer or variable containing an integer). A positive integer constant that gives the AUU/SS rate. The value is in hundredths of an application usage unit (AUU). Thus if the user wanted to charge 2.5 AUUs per SS, IRATE would contain 250.

- ISCHARG (integer or variable containing an integer). A positive integer that is added to the application usage unit (AUU) accumulator. The value is a whole number value which is added as a surcharge to the accumulated AUUs. Thus if the user wished to add 50 AUUs to the AUU accumulator at the termination of a job step, ISCHARG should contain 50.

  If the ISCHARG parameter is specified during a change rate or an end accounting call, ACOUNTX places the AUUs accumulated thus far during the job step in ISCHARG in the form:

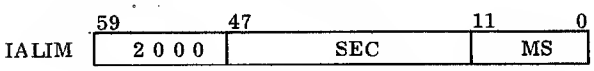


ACC = accumulated AUUs (ACC is in thousandths of AUUs).

  It is for this reason that the input parameter ISCHARG must be a variable rather than a constant or a literal. The returned variable is in thousandths of an AUU. Thus a value of 120505 would indicate that 120.505 AUUs were consumed this far during the job step.

- ISITEM (left justified, blank, or zero fill). A four-character literal used as the item code. This parameter has no meaning. Any literal in this field is displayed in the start message for the application job step in the user dayfile and the system dayfile.

- IALIM (integer or a variable containing an integer). A variable field specified by the calling application used by ACOUNTX to return the number of system seconds (SS) accumulated by the application job. The returned value is a positive integer in thousandths of a system second. Thus a value of 102031 returned to the application as a result of a start or change limit call to ACOUNTX would indicate that 102 seconds and 31 milliseconds were consumed this far. Because IALIM is used by ACOUNTX to return parameters, IALIM must be specified as a variable rather than a constant or a literal value. The returned parameter IALIM takes the form:



  SEC = system seconds accumulated.

  MS = system milliseconds accumulated.

- IVENCD (left justified, blank or zero fill). A two-character literal that is used as the vendor code. This parameter is absolutely required on all start calls to ACOUNTX from non-CDC applications. CDC applications automatically have a vendor code of CD, 00 or ⍽⍽ (⍽=space) assigned by ACOUNTX as would user applications that do not specify an IVENCD parameter on the start call. It is for this reason that user applications calling ACOUNTX choose an IVENCD of two characters other than CD, 00 or ⍽⍽. If the user application supplies one of these three codes as their vendor code, the system will assume Control Data usage pricing liability for the application run.

PARAMETER USAGE

An application job step is defined as the interval between a start and end accounting call. Multiple calls to ACOUNTX to change rate or change limit are permissible after a start and prior to an end call or another start call. Two start calls without an end will cause automatic termination of the first start sequence without any error indication.

Each call to ACOUNTX requires the function to be defined (that is, STAUU, CRAUU, LIMAUU or EDAUU) and from zero to seven parameter specifications depending upon the ACOUNTX call. Table 3-1 defines the required, optional, and parameters not used for each function call. Furthermore, the parameter string may be terminated immediately after the last required parameter of the call.

TABLE 3-1. PARAMETER USAGE TABLE

ICODE	ISFNAME	ISCODE	IRATE	ISCHARG	ISITEM	IALIM	IVENCD*
STAUU	OP	RQ	RQ	NU	OP	OP	RQ*
CRAUU	NU	NU	RQ	RQ	NU	NU	NU
LIMAUU	NU	NU	NU	NU	NU	RQ	NU
EDAUU	NU	NU	NU	OP	NU	NU	NU

Code	Meaning
RQ	Required parameter. Diagnostic is issued if parameter is missing and in some cases if the parameter value is zero or blank.
OP	Optional parameter.
NU	Not used. ACOUNTX does not use these values contained in the parameter nor does it return any values in them.
*	IVENCD is a required parameter only for user supplied applications and an optional parameter for CDC applications. If this parameter is used in CDC supplied applications, the code must be "CD".

PARAMETER FORMATS

The parameter list specification for calls to ACOUNTX are variable length fixed position parameters. Parameters not needed for the particular call may be omitted but the delimiter (,0,) or (,Ø,) must be used to indicate the omission. All parameters must start on a word boundary and agree in type with that shown in table 3-2.

TABLE 3-2. PARAMETER FORMATS

Parameter	Type	FTN	COBOL
ICODE	Literal	5LSTAUU	PIC X(10) VALUE ≠STAUU≠
ICODE	Literal	5LCRAUU	PIC X(10) VALUE ≠CRAUU≠
ICODE	Literal	6LLIMAUU	PIC X(10) VALUE ≠LIMAUU≠
ICODE	Literal	5LEDAUU	PIC X(10) VALUE ≠EDAUU≠
ISFNAME	Literal	10HXXXXXXXXXX	PIC X(10) VALUE ≠XXXXXXXXXX≠
ISCODE	Literal	4LXXXX	PIC X(10) VALUE ≠XXXX≠
IRATE	Integer	Integer Variable	PIC 9(10)
ISCHARG	Integer	Integer Variable	PIC 9(10)
ISITEM	Literal	4LXXXX	PIC X(10) VALUE ≠XXXX≠
IALIM	Integer	Integer Variable	PIC 9(10)
IVENCD	Literal	2LXX	PIC X(10) VALUE ≠XX≠

NOTE

The level number of all COBOL parameters must be level 1 to guarantee that parameters start on a word boundary.

ACOUNTX MESSAGES

During normal processing of accounting calls, ACOUNTX displays start and end messages on the user and system dayfile containing tracking information relative to the job step. (Appendix B contains sample user dayfiles displaying ACOUNTX dayfile messages.)

START MESSAGE

The start message appears on the user and system dayfiles every time a valid start function (STAUU) is received by ACOUNTX. The message takes the following form:

```
hh.mm.ss. SWCS vc swc softwrname icde
```

where:

hh.mm.ss = time of entry (hours/minutes/seconds).

SWCS = indicates a start message.

vc = vendor code (IVENCD parameter) for the application (two characters).

swc = software code (ISCODE parameter) input to ACOUNTX (up to four characters).

softwrname = software product name (ISFNAME parameter) if specified in the call (10 characters).

icde = item code (ISITEM parameter) if specified in the call (four characters).

END MESSAGE

The end message appears on the user and system dayfiles upon termination of an application job step either through an EDAUU call, another start call (STAUU) without a subsequent end call, or an abnormal termination by the system. The end message takes the following form:

```
hh.mm.ss. SWCE vc swc AUUS USED = xxxxxx.xxx
hh.mm.ss.       AUUS ACCUMULATED = tttttt.ttt
```

where:

hh.mm.ss = time of entry (hours/minutes/seconds).

SWCE = indicates an end message.

vc = vendor code (IVENCD parameter) of the application (two characters).

swc = software code (ISCODE parameter) input on the start call to ACOUNTX (up to four characters).

xxxxxx.xxx = the number of AUUs consumed for the job step.

tttttt.ttt = the total number of AUUs consumed thus far for the run.

CALLING ERRORS

ACOUNTX aborts the application if an illegal function request or an illegal parameter is received. An illegal function request results when a function other than STAUU, CRAUU, LIMAUU, or EDAUU is specified as the first parameter in the ACOUNTX calling sequence. Calling errors can also be caused when a required parameter is missing from the parameter list passed to ACOUNTX. A missing parameter can occur when the parameter list is terminated by a zero word prior to the last required parameter for the particular call, in addition, a zero or null parameter is considered illegal for some parameters (that is, ISCODE on start messages and IRATE on start and change rate messages).

When an illegal function or invalid parameter is encountered by ACOUNTX the message:

```
hh.mm.ss. AUU FUNCTION nn, ERR xx
hh.mm.ss. ACOUNT-ILLEGAL PARAMETER (C=aa,P=y)
```

is displayed on the user and system dayfiles and error codes are placed in the detail record for the job step (see the Usage Detail Report subsection in section 2).

where:

nn = a two-character code of from 01 to 06, where:

01 = error on start AUU processing call,

02 = error on change AUU rate call,

03 = error on change AUU limit call,

04 = error on end AUU accounting call,

05 = unknown function input to ACOUNTX, and

06 = force end of software accounting.

and where:

xx = a two-character numeric code of from 01 to 06 indicating error type, where:

01 = illegal software accounting request,

02 = unknown software accounting function,

03 = not used,

04 = not used,

05 = not used, and

06 = illegal parameter.

and where:

aa = a two-character alpha code indicating type of call, where:

ST = STAUU call,

CR = CRAUU call,

CL = LIMAUU call,

ED = EDAUU call, and

UN = unknown function call.

and where:

y = a one-character numeric code indicating parameter error, where:

1 = ICODE parameter in error,

2 = ISFNAME parameter in error,

3 = ISCODE parameter in error,

4 = IRATE parameter in error,

5 = ISCHARG parameter in error,

6 = ISITEM parameter in error,

7 = IALIM parameter in error, and

8 = IVENCD parameter in error.

Calling errors cause AUU accounting to be ended normally with a normal end message (SWCE message) prior to job termination.

ABNORMAL ERRORS

Abnormal errors are any errors other than calling errors which normally would cause a system abort condition. Under these conditions, the application is not given the opportunity to call end accounting to terminate accounting. However, ACOUNTX sets up entries in the system which will force control to ACOUNTX's recovery code so that accounting can be closed out. Table 3-3 defines the error conditions which will initiate ACOUNTX recovery processing.

When an abnormal error condition is detected by the system, control is passed to ACOUNTX which terminates accounting, issues the accounting end (swce) message and banner message if accounting is active, and displays the AUU recovery message to the user and system dayfiles in the following format:

```
hh.mm.ss. AUU RECOVERY ERROR CODE = xx
```

where:

xx = octal error code specified in the recovery error conditions table (table 3-3).

NOTE

For NOS, the exchange package will have registers X1/A1, X6/A6 destroyed.

TABLE 3-3. RECOVERY ERROR CONDITIONS

Error Condition	Error Codes (Octal)	
	NOS/BE	NOS
Normal termination	00	--
Time limit exceeded	01	01
Arithmetic mode error	02	02
PPU requested job abort	03	03
CP program requested abort	04	04
PP cannot be called from RA+1	05	05
Operator DROP	06	06
Operator KILL	07	--
Operator RERUN	10	--
CP abort	11	04
ECS parity error	12	--
System abort	--	12
Required auto-recall status missing	15	05
Job hung in auto-recall	16	--
Required mass storage limit exceeded	17	--
Track limit	--	11
XXX not in PP library	20	05
IO time limit exceeded	21	--

GLOSSARY

ACOUNTX - Application interface which records the number of application usage units (AUU) consumed by an application.

application charge (ISCHARGE) - A positive integer specifying number of AUU's to add to the application usage unit accumulator as a surcharge.

application limit (IALIM) - A variable field specified by the calling application used by ACOUNTX to return the number of system seconds (SS) accumulated by the application job. The returned value is a positive integer in thousandths of system seconds.

application rate (IRATE) - A positive integer constant specifying the number of AUU's/system second for an application job step, in hundreds of AUUs.

application usage unit (AUU) - The base unit used to indicate the amount of usage of an application.

billing run - RANDR run used to generate utilization reports of usage priced applications. Billing runs generate usage totals for up to six months prior to the current month.

Control Data record cards (id=2) - A series of three cards specifying five lines of information, CDC name, address, telephone number and contact (supplied by Control Data from software contract).

customer record cards (id=3) - A series of three cards specifying five lines of customer information to be output on all reports. Information consists of customer name, address, telephone number and contact.

detail usage file - Usage accounting file containing information regarding each run made since the last billing run.

function parameter (ICODE) - A five- or six-character literal defining the function to perform ICODE. Valid parameters are:

STAUU - start accounting,

CRAUU - change AUU rate,

LIMAUU - return the number of system seconds consumed by the application, and

EDAUU - end accounting.

IALIM - Seventh parameter in the parameter string in a call to ACOUNTX (see application limit).

ICODE - First parameter in the parameter string in a call to ACOUNTX (see function parameter).

IRATE - Fourth parameter in the parameter string in a call to ACOUNTX (see application rate).

ISCHARGE - Fifth parameter in the parameter string in a call to ACOUNTX (see application charge).

ISCODE - Third parameter in the parameter string in a call to ACOUNTX (see software code).

ISFNAME - Second parameter in the parameter string in a call to ACOUNTX (see product name).

ISITEM - Sixth parameter in the parameter string in a call to ACOUNTX (see item code descriptor).

item code descriptor (ISITEM) - A four-character literal used to further define the application job step.

IVENCD - Eighth parameter in the parameter string in a call to ACOUNTX (see vendor code).

parameter card (id=1) - A card of the data deck used to specify the Usage Accounting Utility Installation parameters, required on creation run. Data on the card consists of hardware type, number of copies of each report to produce, protection parameters and the type of run (update/billing/summary).

product activity cards (id=4) - This set of data cards defines the valid products and software codes for each product to be usage priced. Information on these cards consists of vendor code for product, product name, valid software codes, the transaction codes (A - for add, D - for delete; A - for active, I - for inactive and threshold protection values). Product activity card information for Control Data products is supplied to the customer in the Application Installation Handbook.

product file - Usage accounting file used as a repository for product name, valid software names, customer/vendor information, monthly AUU totals, and year to date accumulative AUU totals.

product name (ISFNAME) - A literal, one to 10 characters in length, used to designate the software product's name.

software code (ISCODE) - A one- to four-character literal used as the software code to designate a particular job step within a vendor product.

summary run - RANDR run used to generate utilization reports of usage priced applications for the past six months up to the current date.

update run (id=1) - RANDR run used to build and maintain the product file. An update is either specified by a parameter card or implied with address or product cards (id=2, 3 or 4).

update report - Report generated by RANDR as a result of an update listing all the cards processed during the update run and the disposition of each operation.

usage detail report - Report produced as a result of a billing or summary run listing each individual run detail record by software code within product by vendor code.

usage summary report - Report produced as a result of a billing or summary run listing the total liability of usage priced applications by software code within product by vendor code since the last billing run was made and up to a six month history.

vendor code (IVENCD) - A two-character literal used to indicate the vendor of the application. Vendor codes CD, 00 and ⍽⍽ (⍽=space) are reserved for Control Data's use only.

The following three sample programs illustrate a method of calling ACOUNTX from FORTRAN Extended 4, COBOL 4 and COMPASS 3.

All three samples perform the following calls to ACOUNTX with the parameters specified:

Call No. 1 - Start accounting with a rate of 3.5 AUU/SS.

ICODE	= STAUU	- ACCOUNTX function.
ISFNAME	= SAMPLE	- Product name.
ISCODE	= SAMP	- Software code.
IRATE	= 3.5	- AUU rate.
ISCHARG	= Null	- Surcharge.
ISITEM	= blank	- Item code description (not used).
IALIM	= Null	- Limit.
IVENCD	= UU	- Vendor code.

Call No. 2 - Change Rate to 2.0 AUU/SS and add a surcharge of 10 units. The number of accumulated AUU's is returned in ISCHARG.

ICODE	= CRAUU	- ACOUNTX function.
ISFNAME	= Null	- Product name.
ISCODE	= Null	- Software code.
IRATE	= 2.0	- AUU rate.
ISCHARG	= 10	- Surcharge.

Call No. 3 - Call ACOUNTX to determine the number of accumulated system seconds.

ICODE	= LIMAUU	- ACOUNTX function.
ISFNAME	= Null	- Product name.
ISCODE	= Null	- Software code.
IRATE	= Null	- AUU rate.
ISCHARG	= Null	- Surcharge.
ISITEM	= Null	- Item code descriptor.
IALIM	= Any variable	- Limit.

Call No. 4 - Call ACOUNTX to end accounting.

ICODE	= EDAUU	- ACOUNTX function.

Example 1: FTNTEST

FTNTEST is a sample program displaying the method of interfacing ACOUNTX from a FORTRAN-Extended (FTN 4) program.

```
      PROGRAM FTNTEST (OUTPUT)
C        THIS PROGRAM MAKES FOUR CALLS TO ACOUNTX.
C          1. CALL ACOUNTX (5LSTAUU,6LSAMPLE,4LSAMP,350,0,4H    ,0,2HUU)
C          2. CALL ACOUNTX (5HCRAUU,0,0,200,10)
C          3. CALL ACOUNTX (5LLIMAUU,0,0,0,0,ISS)
C          4. CALL ACOUNTX (5LEDAUU)
C        THIS PROGRAM DISPLAYS THE METHOD OF CALLING ACOUNTX BY A
C        PROGRAM WRITTEN IN FTN-4.
      DIMENSION AREA(6),TABL(35)
      DATA AREA /10HNOW IS THE,10H TIME TO C,10HOME TO THE,10H AID OF YO
     C,10HUR COUNTRY,10H . . . ./
   11 FORMAT (10H ISCHARG= ,F10.3)
   22 FORMAT (10H    SS  = ,F10.3)
      J = 1
      K=100
C
C        MAKE START CALL TO ACOUNTX.
C           SOFTWARE CODE = SAMP
C           PRODUCT NAME  = SAMPLE
C           AUU RATE      = 3.5
C           ITEM CODE DES = SAM1
C           VENDOR CODE   = UU
C
      CALL ACOUNTX (6LSTAUU,6LSAMPLE,4LSAMP,350,0,4H    ,0,2HUU)
      CALL FILESQ(TABL,3LLFN,4LABCD,2LFU,2LSQ,2LRT,1LC,2LRI,1LZ,2LFL,60)
      CALL OPENM (TABL,6LOUTPUT)
C        WASTE A LITTLE TIME TO BUILD UP AUUS
      DO 10 I = J,K
      CALL PUT(TABL,AREA,60)
   10 CONTINUE
C        SET UP TO CALL ACOUNTX WITH A RATE CHANGE TO 2.0 AUU/SS
C          AND ADD A SURCHARGE OF 10.
C
      ISCHARG = 10
      CALL ACOUNTX (5HCHAUU,0,0,200,ISCHARG)
C
C        ISCHARG NOW CONTAINS THE NUMBER OF AUUS CONSUMED DURING THIS
C          PORTION OF FTNTEST.
C        ISCHARG = NBR AUU * 1000. SO, DIVIDE BY 1000.
C          PRINT ISCHARG.
C
      CHARG = ISCHARG / 1000.
      PRINT 11, CHARG
C
C        RESET ISCHARG TO ZERO SO AN INADVERTANT CALL TO ACOUNTX WHICH
C          WOULD ADD A SURCHARGE TO THE AUUS. THE SURCHARGE WILL BE 0.
C
C
C        WASTE MORE TIME TO ACCUMULATE MORE AUUS.
C
      DO 20 I = 1,20
      CALL PUT (TABL,AREA,60)
   20 CONTINUE
C
C        MAKE CALL TO ACOUNTX TO OBTAIN THE NUMBER OF SYSTEM SECONDS
C        CONSUMED THUS FAR.
C
      ISS = 50
      CALL ACOUNTX (6LLIMAUU,0,0,0,0,ISS)
C
C        ISS CONTAINS THE NUMBER OF SS CONSUMED * 1000.  SO DIVIDE BY
```

```
      65          C          1000.
                  C
                        SS = ISS /1000.
                        PRINT 22, SS
                  C
      70          C        NOW WE CAN COAST ALL THE WAY TO OUR END CALL.  LET'S ACCUMULATE
                  C          MORE AUUS.
                  C
                        DO 30 I = 1,20
                        CALL PUT (TABL,AREA,60)
      75           30   CONTINUE
                  C
                  C        WE ARE NOT GOING TO SURCHARG THIS END CALL SO WE DON'T HAVE
                  C          TO MAKE SURE THAT ISCHARG IS SET TO THE VALUE WE WANT TO
                  C          SURCHARGE THE RUN, BUT IF WE DID WE WOULD SET ISCHARG TO
      80          C          THE PROPER VALUE, SAY 20, AND CALL ACOUNTX BY THE FOLLOWINGS
                  C               ISCHARG = 20
                  C               CALL ACOUNTX (SLEDAUU,0,0,0,ISCHARG)
                  C        BUT WE WILL MAKE OUR CALL WITHOUT ISCHARG.
                  C
      85                  CALL ACOUNTX (SLEDAUU)
                  C
                  C          TIME TO CLOSE THE FILE AND GO HOME.
                  C
                        CALL CLOSEM (TABL)
      90                END
```

```
00.13.19.FTN002M  FROM
00.13.19.IP  0000576 WORDS - FILE INPUT  , DC 60
00.13.19.FTN,CM50000,T60.
00.13.19.FTN.
00.13.23.       .509 CP SECONDS COMPILATION TIME
00.13.23.LGO.
00.13.30.  SWCS  UU  SAMP  SAMPLE
00.13.30.  SWCE  UU  SAMP  AUUS USED =      10.244
00.13.30.            AUUS ACCUMULATED =     10.244
00.13.31.      END  FINIEST
00.13.31.       .164 CP SECONDS EXECUTION TIME
00.13.31.OP  0000544U WORDS - FILE OUTPUT , DC 60
00.13.31.MS     7168  WORDS  (        0 MAX USED)
00.13.31.CPA       5.266 SEC.         5.266 ADJ.
00.13.31.IO        1.109 SEC.         1.109 ADJ.
00.13.31.CM      130.408 KWS.         7.959 ADJ.
00.13.31.SS                          14.335
00.13.31.PP        6.661 SEC.
00.13.31.EJ  END OF JOB, **

**********          FTN002M  //// END OF LIST ////
**********          FTN002M  //// END OF LIST ////
```

Example 2: COBTEST

COBTEST is a sample program displaying the method of interfacing ACOUNTX from a COBOL 4 program.

```
00001          IDENTIFICATION DIVISION.
00002              PROGRAM-ID. COBTEST.
00003              AUTHOR. COC.
00004              INSTALLATION.
00005              DATE-WRITTEN.
00006                    SEPTEMBER 9, 1975.
00007          *   REMARKS.
00008          *       -THIS PROGRAM WILL BE USED TO SIMULATE A USER PROGRAM
00009          *        USING PARAMETERS NEEDED TO FUNCTION ACOUNTX IN A COBOL
00010          *        ENVIRONMENT.
00011          *       -FOUR CALLS WILL BE MADE TO ACOUNTX
00012          *          ONE      STAUU WITH IRATE EQUAL TO 3.5
00013          *          TWO      CRAUU WITH IRATE EQUAL TO 2 AND ISCHARG = 10
00014          *          THREE    LIMUU TO GET NUMBER OF SS USED
00015          *          FOUR     EDAUU WITH NO OTHER PARAMETERS.
00016          *       -THIS PROGRAM DISPLAYS THE METHOD OF CALLING ACOUNTX BY
00017          *        A COBOL PROGRAM.
00018          ENVIRONMENT DIVISION.
00019          CONFIGURATION SECTION.
00020          SOURCE-COMPUTER.
00021                    6000.
00022          OBJECT-COMPUTER.
00023                    6000.
00024          INPUT-OUTPUT SECTION.
00025          FILE-CONTROL.
00026                SELECT CARDS ASSIGN TO INPUT.
00027          DATA DIVISION.
00028          WORKING-STORAGE SECTION.
00029              01 ICODE                        PIC X(10) VALUE #STAUU#.
00030              01 ISFNAME                      PIC X(10) VALUE #SAMPLE#.
00031              01 ISCODE                       PIC X(10) VALUE #SAMP#.
00032              01 IRATE     USAGE IS COMP-1    PIC 9(10) VALUE 350.
00033              01 ISCHARG   USAGE IS COMP-1    PIC 9(10) VALUE 10.
00034              01 ISITEM                       PIC X(10) VALUE #    #.
00035              01 IALIM     USAGE IS COMP-1    PIC 9(10) VALUE ZEROS.
00036              01 IVENCD                       PIC X(10) VALUE #00#.
00037              01 COUNT-A                      PIC 9(10) VALUE ZEROS.
00038              01 II        USAGE IS COMP-1    PIC 9(10).
00039          PROCEDURE DIVISION.
00040          FUNCTION-START.
00041                ENTER FORTRAN-X ACOUNTX USING ICODE, ISFNAME, ISCODE,
00042                 IRATE, ISCHARG, ISITEM, IALIM, IVENCD.
00043                PERFORM TIME-ROUT THRU TIME-EXIT VARYING II FROM 1 BY 1
00044                 UNTIL II GREATER THAN 5000.
00045          FUNCTION-CHANGE-RATE.
00046                MOVE #CRAUU# TO ICODE.
00047                MOVE 200 TO IRATE.
00048                ENTER FORTRAN-X ACOUNTX USING ICODE, ISFNAME, ISCODE,
00049                 IRATE, ISCHARG.
00050                PERFORM TIME-ROUT THRU TIME-EXIT VARYING II FROM 1 BY 1
00051                 UNTIL II GREATER THAN 50000.
00052          FUNCTION-CHANGE-LIMIT.
00053                MOVE #LIMAUU# TO ICODE.
00054                MOVE ZEROS TO IALIM.
00055                ENTER FORTRAN-X ACOUNTX USING ICODE, ISFNAME, ISCODE,
00056                 IRATE, ISCHARG, ISITEM, IALIM.
00057          FUNCTION-END-ACCOUNTING.
00058                PERFORM TIME-ROUT THRU TIME-EXIT VARYING II FROM 1 BY 1
00059                 UNTIL II GREATER THAN 70000.
00060                MOVE #EDAUU# TO ICODE.
```

```
00061                   MOVE ZERO TO ASCHARG.
00062                   MOVE *EDAKU* TO ICODE.
00063                   ENTER FORTRAN-X ACOUNTX USING ICODE, TSFNAME, ISCODE,
00064                    IRATE, ISCHARG, ISITEM, TALIM.
00065           TIME-ROUT.
00066                   ADD 1 TO COUNT-A.
00067           TIME-EXIT.
00068                   MOVE ZERO TO COUNT-A.
00069           END-PROGRAM.
00070                   STOP RUN.
```

```
00.12.49.COBTE2K  FRUM
00.12.49.IP  00000640 WORDS - FILE INPUT  , DC 50
00.12.49.COBTEST,CM65000,T200.
00.12.49.COBOL.
00.12.51.COMPILING COBTEST
00.12.55.  000 E AND     T/U  DIAGNOSTICS ISSUED
00.12.55.     054400B SCM USED
00.12.55.       .924 CP SECONDS COMPILATION TIME
00.12.55.END COBOL
00.12.56.MAP,OFF.
00.12.56.LGO.
00.13.02.  SWCS  UU  SAMP  SAMPLE
00.13.15.  SWCE  UU  SAMP  AUUS USED =      86.954
00.13.15.            AUUS ACCUMULATED =     86.954
00.13.16.OP  00001216 WORDS - FILE OUTPUT , DC 40
00.13.16.MS     3584  WORDS  (        0 MAX USED)
00.13.16.CPA      13.319 SEC.        13.319 ADJ.
00.13.16.IO        1.221 SEC.         1.221 ADJ.
00.13.16.CM      226.830 KWS.        13.844 ADJ.
00.13.16.SS                          28.385
00.13.16.PP        9.107 SEC.      DATE 08/01/75
00.13.16.EJ  END OF JOB, **
```

Example 3: CMPTEST

CMPTEST is a sample program displaying the method of interfacing ACOUNTX from a COMPASS 3 program.

```
                                        IDENT  CMPTEST
                                        ENTRY  BEGIN
                                        EXT    ACOUNTX
                                *
                                *       THIS PROGRAM DISPLAYS A METHOD OF CALLING ACOUNTX BY A
                                *       PROGRAM WRITTEN IN COMPASS VERSION 3.
                                *
                                *       PARAMETER ADDRESS BLOCK.  TERMINATED BY A WORD OF ZEROS
                                *
                                *
 0  03152024052324              TRACE  VFD    42/1HCMPTEST
                  000022 *             VFD    18/BEGIN
                                *
 1  000000000000000000012 *     PARADD VFD    60/1CODE    - FUNCTION
 2  000000000000000000013 *            VFD    60/1SFNAME  - PRODUCT NAME
 3  000000000000000000014 *            VFD    60/1SCODE   - SOFTWARE CODE
 4  000000000000000000015 *            VFD    60/1RATE    - AUU RATE / SYSTEM SECOND
 5  000000000000000000016 *            VFD    60/1SCHARG  - SURCHARGE
 6  000000000000000000017 *            VFD    60/1SITEM   - ITEM CODE DESCRIPTOR
 7  000000000000000000020 *            VFD    60/1ALIM    - LIMIT.  RETURN OF ACCUM SS.
10  000000000000000000021 *            VFD    60/1VENCD   - VENDOR CODE
11                     1               BSSZ   1          - ZERO TERMINATOR FOR PARAMETER ADDR LIST
                                *
                                *       PARAMETER AREA.
                                *
12  232401252S0000000000        1CODE  VFD    60/5LSTAUU
13  230115240S0000000000        1SFNAME VFD    60/6LSAMPLE
14  230115200000000000000       1SCODE VFD    60/4LSAMP
15  000000000000000000536       1RATE  DATA   3500
16  000000000000000000000       1SCHARG DATA  0
17  555555550000000000000       1SITEM VFD    60/4L
20  000000000000000000000       1ALIM  DATA   0
21  252S000000000000000000      1VENCD VFD    60/2LUU
                                *
22                              BEGIN  BSS    0
                                *
                                *       MAKE START CALL
                                *
22  S1100000001 *                      SA1    PARADD      (A1) = ADDR OF PARAM ADDR BLOCK
23  0100000000 X                *      RJ     ACOUNTX     START ACCOUNTING
                  0023 *        -      VFD    12/*,18/TRACE      (TRACE WORD + LINE NUMBER)
                     000000 *
                                *
                                *       WASTE SOME TIME TO ACCUMULATE AUUS.
                                *
24  6110000001                         SB1    1
                        6120017777     SB2    17777B
25  67221                       CMP.1  SB2    B2-B1
                  052000002S *         NE     B2,B0,CMP.1
26  7160000012                         SX6    100         SETUP FOR CALL TO CHANGE RATE
                        7170000310     SX7    2000        NEW RATE = 2.0
27  S1200000S1 *                       SA2    =H*CRAUU*   CHANGE FUNCTION TO CHANGE RATE
                        5160000016 *   SA6    1SCHARG     SET SURCHARGE IN 1SCHARG PARAMETER
30  S1700000015 *                      SA7    1RATE       SET NEW RATE IN 1RATE
                        10622          BX6    X2          SET CRAUU IN 1CODE
31  S1100000001 *                      SA1    PARADD      (A1) = ADDR OF PARAM ADDR BLOCK
                        S1600000012 *  SA6    1CODE
32  0100000000 X                *      RJ     ACOUNTX     CHANGE RATE
                  0032 *        -      VFD    12/*,18/TRACE      (LINE NBR + TRACE WORD)
                     000000 *
```

```
                                        *
                                        *       THE ELEMENT ISCHARG WILL CONTAIN THE NUMBER OF AUU * 1000
                                        *       UPON RETURN FROM THE CHANGE RATE CALL.  ISCHARG MUST BE
                                        *       SET TO THE NEW SURCHARGE RATE IF SUBSEQUENT CALL TO
                                        *       CHANGE RATE OR END (WITH ISCHARG SPECIFIED) ARE MADE.  IF
                                        *       IT IS NOT RESET AND A CALL TO ACOUNTX IS MADE WHICH SPECIFIES
                                        *       ISCHARG AS A REQUIRED OR OPTIONAL PARAMETER, THE RETURNED
                                        *       VALUE FROM THE PREVIOUS CALL WILL BE USED AS THE SURCHARGE.
                                        *       THIS VALUE IS MULTIPLIED BY 1000 BEFORE IT IS RETURNED TO
                                        *       THE CALLER, SO THAT WOULD BE A GOOD WAY TO ACCUMULATE CHARGES
                                        *       VERY RAPIDLY.  WE WILL CLEAR ISCHARG NOW SO WE DON'T FORGET.
                                        *
          33  6120007777                            SB2     7777B       WASTE MORE TIME TO BUILD UP AUUS AND ES
          34  67221                       CMP.S     SB2     B2-B1
                    0520000034 +                    NE      B2,B0,CMP.S
                                        *       IALIM WAS CHANGED FROM 50 TO A NEW VALUE,  THE NUMBER
                                        *       OF SS ACCUMULATED BY THE START CALL SO IALIM MUST
                                        *       BE CHANGED TO THE NEW VALUE OF SS LIMIT.
          35  7160000106                            SX6     100
                         5120000052 +               SA2     =H*LIMAUU*
          36  10722                                 BX7     X2
                    5160000020 +                    SA6     IALIM
          37  S1700000012 +                         SA7     ICODE
                         5110000001 +               SA1     PARAOO      IALIM SO NO ERROR OCCURS. GET PARAM LIST AD
          40  0100000000 X                 +         RJ      ACOUNTX     MAKE LIMIT CALL
                         0040 +               -      VFD     12/*,18/TRACE
                              000000 +
                                        *
                                        *       UPON RETURN, THE ELEMENT IALIM WILL CONTAIN THE NUMBER OF
                                        *       SYSTEM SECONDS ACCUMULATED THUS FAR.  IT IS A GOOD PRACTICE
                                        *       TO CLEAR THIS ELEMENT AFTER IT IS NOT NEEDED.
                                        *
          41  43700                                 MX7     0
                    6120017777                      SB2     17777B      WASTE MORE TIME.  THEN MAKE END CALL.
          42  67221                       CMP.10    SB2     B2-B1
                    0520000042 +                    NE      B2,B0,CMP.10
          43  S1700000020 +                         SA7     IALIM       RESET IALIM
                                        *
                                        *       SETUP TO END AUU ACOUNTING.  OUR CALL WILL CONSIST OF
                                        *       JUST THE ICODE PARAMETER NO! THE ISCHARG PARAMETER AS WELL.
                                        *       THUS WE SET THE ADDRESS LIST TERMINATOR IMMEDIATELY FOLLOWING
                                        *       THE ICODE PARAMETER IN THE ADDRESS LIST.
                                        *
                         5120000053 +               SA2     =H*EOAUU*
          44  S1700000002 +                         SA7     PARADD+1
                         10722                      BX7     X2
          45  S1700000012 +                         SA7     ICODE
                         5110000001 +               SA1     PARADD      GET PARAMETER ADDR LIST ADDRESS.
          46  0100000000 X                 +         RJ      ACOUNTX     MAKE END CALL
                         0046 +               -      VFD     12/*,18/TRACE
                              000000 +
                                        *
                                        *       WE ARE FINISHED, YOU CAN GO HOME NOW.
                                        *
          47  7160247021                            ENDRUN
          54                                        END     BEGIN


00.13.01.CMPTE2L  FROM
00.13.01.IP  0000076B WORDS - FILE INPUT  , DC 60
00.13.01.CMPTEST,CMS0000,T20.
00.13.02.COMPASS.
00.13.05. ASSEMBLY COMPLETE.   46400B  CM USED.
00.13.05.     1.832  CPU SECONDS  ASSEMBLY TIME.
00.13.06.LGO.
00.13.12.  SWCS  UU  SAMP  SAMPLE
00.13.12.  SWCE  UU  SAMP  AUUS USED =      10.070
00.13.12.            AUUS ACCUMULATED =     10.070
00.13.12.OP  00004352 WORDS - FILE OUTPUT , DC 40
00.13.12.MS    7168  WORDS  (        0 MAX USED)
00.13.12.CPA       4.555 SEC.          4.555 ADJ.
00.13.12.IO         .027 SEC.           .027 ADJ.
00.13.12.CM      105.988 KWS.          6.468 ADJ.
00.13.13.SS                           11.052
00.13.13.PP        4.708 SEC.      DATE 08/01/75
00.13.13.EJ  END OF JOB, **
```

CPU MODEL CODES

C

Upon creation of the product file, the parameter card (id=1) must be specified with UPDATE as the run type and a machine code (CPU model code). The appropriate code is to be chosen from the table below:

Machine	Machine Code
6200	CY62
6400	CY64
6500	CY65
6600	CY66
6700	CY67
CYBER 71	CY71
CYBER 72	CY72
CYBER 73	CY73
CYBER 74	CY74
CYBER 171	C171
CYBER 172	C172
CYBER 173	C173
CYBER 174	C174
CYBER 175	C175
CYBER 175-1	L175
CYBER 175-2	C175
CYBER 175-3	U175
CYBER 176	C176

The customer is obligated, in the event of a CPU model upgrade, [for example, a CYBER 172 (C172) with a 10316-1 upgrade is a CYBER 173 (C173)], to perform a final billing run on existing product and usage detail files, remove those files and recreate them with the appropriate CPU model code.

COMMENT SHEET

MANUAL TITLE Usage Accounting Utility Reference Manual

PUBLICATION NO. 84000440 REVISION B

FROM: NAME:

BUSINESS ADDRESS:

COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual. Customer engineers are urged to use the TAR.

CUT ALONG LINE

AA3419 REV. 11/69 PRINTED IN U.S.A

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Publications and Graphics Division
P. O. Box 0 HQC02C
Minneapolis, Minnesota 55440

ATTN: DATA SERVICES PUBLICATIONS

CUT ALONG LINE

FOLD

FOLD